



Open Source Used In AppDynamics_OTIS_Pipeline 22.12.722

Cisco Systems, Inc.

www.cisco.com

Cisco has more than 200 offices worldwide.
Addresses, phone numbers, and fax numbers
are listed on the Cisco website at
www.cisco.com/go/offices.

Text Part Number: 78EE117C99-1525247838

This document contains licenses and notices for open source software used in this product. With respect to the free/open source software listed in this document, if you have any questions or wish to receive a copy of any source code to which you may be entitled under the applicable free/open source license(s) (such as the GNU Lesser/General Public License), please submit this [form](#).

In your requests please include the following reference number 78EE117C99-1525247838

Contents

1.1 prometheus-procfs v0.7.3

1.1.1 Available under license

1.2 error_prone_annotations 2.7.1

1.2.1 Available under license

1.3 x-sys v0.1.0

1.3.1 Available under license

1.4 jmes-path-query-library 1.12.349

1.4.1 Available under license

1.5 amazon-kinesis-client-library-for-java 2.4.2

1.5.1 Available under license

1.6 cespare-xxhash v1.1.0

1.6.1 Available under license

1.7 jacoco 0.8.7

1.7.1 Available under license

1.8 kotlin-script-runtime 1.7.20

1.8.1 Available under license

1.9 apache-avro 1.11.1

1.9.1 Available under license

1.10 ginkgo 1.16.4

1.10.1 Available under license

1.11 docker-java-transport 3.2.13

1.11.1 Available under license

1.12 jetbrains-annotations 13.0

1.12.1 Available under license

1.13 golang-appengine v1.6.7

1.13.1 Available under license

1.14 joda-time v2.10.2

1.14.1 Available under license

1.15 lz4-and-xxhash 1.7.1

1.15.1 Available under license

1.16 github.com/xiaomi/pegasus-go-client 20210427-snapshot-f3b6b08b

1.16.1 Available under license

1.17 text v0.4.0

1.17.1 Available under license

1.18 kr-pretty 0.1.0

1.18.1 Available under license

1.19 aws-java-sdk-:::core-:::protocols-:::json-utils 2.17.122

1.19.1 Available under license

1.20 junit-platform-junit-platform-engine 1.8.2

1.20.1 Available under license

1.21 testcontainers-core 1.17.3

1.21.1 Available under license

1.22 apache-http-client 4.5.13

1.22.1 Available under license

1.23 zstd-jni 1.4.9-1

1.23.1 Available under license

1.24 apache-log4j 2.17.1

1.24.1 Available under license

1.25 netty-reactive-streams-implementation 2.0.5

1.25.1 Available under license

1.26 wire-schema 3.7.1

1.26.1 Available under license

1.27 aws-sdk-for-go v1.44.140

1.27.1 Available under license

1.28 wire-protocol-buffer-java-generator 3.7.1

1.28.1 Available under license

1.29 go-tomb-tomb 20150422-snapshot-dd632973

1.29.1 Available under license

1.30 spf13-cast v1.4.1

1.30.1 Available under license

1.31 aws-sdk-for-java 1.12.349

1.31.1 Available under license

1.32 opencensus-api 0.28.0

1.32.1 Available under license

1.33 asm-analysis 9.1

- 1.33.1 Available under license
- 1.34 slf4j-api-module 1.7.32**
 - 1.34.1 Available under license
- 1.35 rxjava 3.1.3**
 - 1.35.1 Available under license
- 1.36 kotlin 1.7.20**
 - 1.36.1 Available under license
- 1.37 project-lombok 1.18.20**
 - 1.37.1 Available under license
- 1.38 thanhpk/randstr v1.0.4**
 - 1.38.1 Available under license
- 1.39 aws-sdk-for-java 2.17.122**
 - 1.39.1 Available under license
- 1.40 smartystreets-assertions v1.2.1**
 - 1.40.1 Available under license
- 1.41 cocood-freecache v1.2.0**
 - 1.41.1 Available under license
- 1.42 guava-listenablefuture-only 9999.0-empty-to-avoid-conflict-with-guava**
 - 1.42.1 Available under license
- 1.43 re2j 1.5**
 - 1.43.1 Available under license
- 1.44 pkgerrors v0.9.1**
 - 1.44.1 Available under license
- 1.45 scala v2.12.10**
 - 1.45.1 Available under license
- 1.46 jetbrains-kotlin-kotlin-stdlib-jdk7 1.7.20**
 - 1.46.1 Available under license
- 1.47 everit-json-schema 1.12.2**
 - 1.47.1 Available under license
- 1.48 netty/resolver/dns/classes/macos 4.1.85.Final**
 - 1.48.1 Available under license
- 1.49 junit 4.13.2**
 - 1.49.1 Available under license
- 1.50 grpc-netty-shaded 1.39.0**
 - 1.50.1 Available under license
- 1.51 apache-commons-digester 2.1**
 - 1.51.1 Available under license
- 1.52 bean-validation-api 2.0.1**
 - 1.52.1 Available under license

1.53 goconvey v1.7.2

1.53.1 Available under license

1.54 mockito-junit-jupiter 3.11.2

1.54.1 Available under license

1.55 duct-tape 1.0.8

1.55.1 Available under license

1.56 leaktest v1.3.0

1.56.1 Available under license

1.57 json-java 20201115

1.57.1 Available under license

1.58 kotlin-reflect 1.7.20

1.58.1 Available under license

1.59 prometheus-common v0.32.1

1.59.1 Available under license

1.60 javax-annotation-api 1.3.2

1.60.1 Available under license

1.61 google-gson 2.10

1.61.1 Available under license

1.62 kubernetes-apimachinery 20191123-snapshot-4c4803ed

1.62.1 Available under license

1.63 aws-java-sdk-::-services-::-amazon-dynamodb 2.17.101

1.63.1 Available under license

1.64 aws-sdk-for-java 2.17.101

1.64.1 Available under license

1.65 kotlin-scripting-common 1.7.20

1.65.1 Available under license

1.66 aws-java-sdk-::-services-::-aws-glue 2.17.122

1.66.1 Available under license

1.67 junit-jupiter-junit-jupiter-api 5.8.2

1.67.1 Available under license

1.68 jackson-dataformats-binary 2.14.0

1.68.1 Available under license

1.69 apache-httpcomponents-core 4.4.13

1.69.1 Available under license

1.70 handy-uri-templates handy-uri-templates-2.1.8

1.70.1 Available under license

1.71 jetbrains-kotlin-kotlin-scripting-compiler-embeddable 1.3.50

1.71.1 Available under license

1.72 jmespath-go-jmespath v0.4.0

- 1.72.1 Available under license
- 1.73 jackson-annotations 2.14.0**
 - 1.73.1 Available under license
- 1.74 snakeyaml-engine 2.5**
 - 1.74.1 Available under license
- 1.75 assertj-fluent-assertions 3.22.0**
 - 1.75.1 Available under license
- 1.76 jetbrains-kotlin-kotlin-stdlib-jdk8 1.7.20**
 - 1.76.1 Available under license
- 1.77 aws-java-sdk-::third-party-::jackson-dataformat-cbor 2.17.101**
 - 1.77.1 Available under license
- 1.78 jackson-core 2.14.0**
 - 1.78.1 Available under license
- 1.79 re2j 1.3**
 - 1.79.1 Available under license
- 1.80 byte-buddy byte-buddy-1.11.3**
 - 1.80.1 Available under license
- 1.81 slf4j-api-module 1.7.36**
 - 1.81.1 Available under license
- 1.82 kotlinx-coroutines-core 1.1.1**
 - 1.82.1 Available under license
- 1.83 google-guava 31.0.1-jre**
 - 1.83.1 Available under license
- 1.84 junit-jupiter-junit-jupiter-params 5.8.2**
 - 1.84.1 Available under license
- 1.85 netty-reactive-streams-http-support 2.0.5**
 - 1.85.1 Available under license
- 1.86 go-testify 1.7.0**
 - 1.86.1 Available under license
- 1.87 pegasus-kv/thrift v0.13.0**
 - 1.87.1 Available under license
- 1.88 findbugs-jsr305 3.0.2**
 - 1.88.1 Available under license
- 1.89 aws-java-sdk-::core-::-protocols-::-aws-cbor-protocol 2.17.101**
 - 1.89.1 Available under license
- 1.90 kotlin-scripting-compiler-impl-embeddable 1.3.50**
 - 1.90.1 Available under license
- 1.91 apache-commons-validator 1.7**
 - 1.91.1 Available under license

- 1.92 opentest4j-opentest4j 1.2.0**
 - 1.92.1 Available under license
- 1.93 tail v1.4.8**
 - 1.93.1 Available under license
- 1.94 aws-java-sdk-::-services-::-amazon-kinesis 2.17.101**
 - 1.94.1 Available under license
- 1.95 classgraph 4.8.120**
 - 1.95.1 Available under license
- 1.96 pgv-java-stubs 0.6.1**
 - 1.96.1 Available under license
- 1.97 go-humanize v1.0.0**
 - 1.97.1 Available under license
- 1.98 protobuf v1.27.1**
 - 1.98.1 Available under license
- 1.99 commons-logging 1.2**
 - 1.99.1 Available under license
- 1.100 x-xerrors 20200804-snapshot-5ec99f83**
 - 1.100.1 Available under license
- 1.101 aws-secretsmanager-caching-java 1.0.1**
 - 1.101.1 Available under license
- 1.102 perfmark-perfmark-api 0.23.0**
 - 1.102.1 Available under license
- 1.103 j2objc-annotations 1.3**
 - 1.103.1 Available under license
- 1.104 wire-protocol-buffer-runtime 3.7.1**
 - 1.104.1 Available under license
- 1.105 apache-log4j-slf4j-binding 2.19.0**
 - 1.105.1 Available under license
- 1.106 golang-mock v1.6.0**
 - 1.106.1 Available under license
- 1.107 io-grpc-grpc-context 1.39.0**
 - 1.107.1 Available under license
- 1.108 aws-glue-schema-registry-build-tools 1.1.9**
 - 1.108.1 Available under license
- 1.109 netty-project 4.1.68.Final**
 - 1.109.1 Available under license
- 1.110 gomemcache 20190913-snapshot-a41fca85**
 - 1.110.1 Available under license
- 1.111 fsnotify-fsnotify v1.4.9**

- 1.111.1 Available under license
- 1.112 software-amazon-ion-ion-java 1.0.2**
 - 1.112.1 Available under license
- 1.113 pgv-java-grpc-interceptors 0.6.1**
 - 1.113.1 Available under license
- 1.114 testcontainers-::localstack 1.17.3**
 - 1.114.1 Available under license
- 1.115 caffeine-cache 3.0.3**
 - 1.115.1 Available under license
- 1.116 aws-java-sdk-::core-::-protocols-::-aws-json-protocol 2.17.122**
 - 1.116.1 Available under license
- 1.117 asm 9.1**
 - 1.117.1 Available under license
- 1.118 aws-event-stream 1.0.1**
 - 1.118.1 Available under license
- 1.119 apache-log4j-api 2.17.1**
 - 1.119.1 Available under license
- 1.120 animal-sniffer-annotation 1.19**
 - 1.120.1 Available under license
- 1.121 beorn7-perks v1.0.1**
 - 1.121.1 Available under license
- 1.122 protobuf-java 3.19.4**
 - 1.122.1 Available under license
- 1.123 checker-qual 3.15.0**
 - 1.123.1 Available under license
- 1.124 netty-codec-smtp 4.1.85.Final**
 - 1.124.1 Available under license
- 1.125 apache-kafka 2.8.2**
 - 1.125.1 Available under license
- 1.126 x-sync 20220722-snapshot-886fb937**
 - 1.126.1 Available under license
- 1.127 jetbrains-annotations 17.0.0**
 - 1.127.1 Available under license
- 1.128 javapoet 1.13.0**
 - 1.128.1 Available under license
- 1.129 aws-glue-schema-registry-avro-serializer-deserializer 1.1.9**
 - 1.129.1 Available under license
- 1.130 commons-compress 1.21**
 - 1.130.1 Available under license

- 1.131 google-guava 30.1-android**
 - 1.131.1 Available under license
- 1.132 kotlin-libraries-bill-of-materials 1.7.20**
 - 1.132.1 Available under license
- 1.133 netty/transport/classes/epoll 4.1.85.Final**
 - 1.133.1 Available under license
- 1.134 docker-java 3.2.13**
 - 1.134.1 Available under license
- 1.135 kotlin-scripting-compiler-impl-embeddable 1.7.20**
 - 1.135.1 Available under license
- 1.136 wire 3.7.1**
 - 1.136.1 Available under license
- 1.137 golang-glog 20160125-snapshot-23def4e6**
 - 1.137.1 Available under license
- 1.138 prometheus-client-model v0.2.0**
 - 1.138.1 Available under license
- 1.139 go-logrus v1.8.1**
 - 1.139.1 Available under license
- 1.140 gomega v1.16.0**
 - 1.140.1 Available under license
- 1.141 google-gson 2.8.6**
 - 1.141.1 Available under license
- 1.142 jtolds-gls v4.20.0**
 - 1.142.1 Available under license
- 1.143 apache-log4j 2.19.0**
 - 1.143.1 Available under license
- 1.144 google-uuid v.1.1.2**
 - 1.144.1 Available under license
- 1.145 jetbrains-kotlin-kotlin-stdlib-jdk8 1.4.10**
 - 1.145.1 Available under license
- 1.146 byte-buddy-agent 1.11.3**
 - 1.146.1 Available under license
- 1.147 asm-commons 9.1**
 - 1.147.1 Available under license
- 1.148 kotlinpoet 1.7.2**
 - 1.148.1 Available under license
- 1.149 jna 5.8.0**
 - 1.149.1 Available under license
- 1.150 kotlinx-serialization-core 1.4.0**

- 1.150.1 Available under license
- 1.151 aws-java-sdk::-http-clients::-url-connection 2.17.122**
 - 1.151.1 Available under license
- 1.152 vmihailenco/msgpack v4.0.4**
 - 1.152.1 Available under license
- 1.153 jetbrains-kotlin-kotlin-scripting-compiler-embeddable 1.7.20**
 - 1.153.1 Available under license
- 1.154 netty/transport/classes/kqueue 4.1.85.Final**
 - 1.154.1 Available under license
- 1.155 snappy-java 1.1.8.1**
 - 1.155.1 Available under license
- 1.156 kotlin-stdlib-common 1.7.20**
 - 1.156.1 Available under license
- 1.157 junit 4.12**
 - 1.157.1 Available under license
- 1.158 error_prone_annotations 2.3.4**
 - 1.158.1 Available under license
- 1.159 wire-protocol-buffer-compiler 3.7.1**
 - 1.159.1 Available under license
- 1.160 burtsushi-toml v0.3.1**
 - 1.160.1 Available under license
- 1.161 apache-commons-collections 3.2.2**
 - 1.161.1 Available under license
- 1.162 aws-java-sdk::-services::-amazon-cloudwatch 2.17.101**
 - 1.162.1 Available under license
- 1.163 x-net v0.1.0**
 - 1.163.1 Available under license
- 1.164 allegro-bigcache v2.2.5**
 - 1.164.1 Available under license
- 1.165 go-check-check 20190902-snapshot-41f04d3b**
 - 1.165.1 Available under license
- 1.166 project-lombok 1.18.12**
 - 1.166.1 Available under license
- 1.167 aws-java-sdk::-metrics-interface 2.17.122**
 - 1.167.1 Available under license
- 1.168 dgryski/go-rendezvous 20200823-snapshot-9f7001d1**
 - 1.168.1 Available under license
- 1.169 dgraph-io/ristretto v0.1.0**
 - 1.169.1 Available under license

- 1.170 reactive-streams v1.0.3**
 - 1.170.1 Available under license
- 1.171 netty-transport-native-unix-common 4.1.85.Final**
 - 1.171.1 Available under license
- 1.172 checker-qual 2.5.5**
 - 1.172.1 Available under license
- 1.173 gopherjs 20181205-snapshot-0766667c**
 - 1.173.1 Available under license
- 1.174 docker-java-transport-zero-dep 3.2.13**
 - 1.174.1 Available under license
- 1.175 protobuf-java 3.21.9**
 - 1.175.1 Available under license
- 1.176 mockito v3.11.2**
 - 1.176.1 Available under license
- 1.177 patrickmn-go-cache 2.1.0**
 - 1.177.1 Available under license
- 1.178 aws-java-sdk-::third-party-::jackson-core 2.17.122**
 - 1.178.1 Available under license
- 1.179 aws-java-sdk-::arns 2.17.122**
 - 1.179.1 Available under license
- 1.180 io-grpc-grpc-api 1.39.0**
 - 1.180.1 Available under license
- 1.181 mac-os 4.1.85.Final**
 - 1.181.1 Available under license
- 1.182 yaml-for-go v2.4.0**
 - 1.182.1 Available under license
- 1.183 protocol-buffer-java-util-package 3.15.6**
 - 1.183.1 Available under license
- 1.184 cespare-xxhash v2.1.2**
 - 1.184.1 Available under license
- 1.185 google-api-grpc-proto-google-common-protos 2.7.4**
 - 1.185.1 Available under license
- 1.186 jackson-databind 2.14.0**
 - 1.186.1 Available under license
- 1.187 apache-commons-lang 3.12.0**
 - 1.187.1 Available under license
- 1.188 google-android-annotations-library 4.1.1.4**
 - 1.188.1 Available under license
- 1.189 asm-tree 9.1**

- 1.189.1 Available under license
- 1.190 kaml 0.49.0**
 - 1.190.1 Available under license
- 1.191 junit-jupiter-aggregator 5.8.2**
 - 1.191.1 Available under license
- 1.192 commons-codec 1.15**
 - 1.192.1 Available under license
- 1.193 kqueue 4.1.85.Final**
 - 1.193.1 Available under license
- 1.194 apache-commons-validator 1.6**
 - 1.194.1 Available under license
- 1.195 aws-java-sdk-secretsmanager 1.11.409**
 - 1.195.1 Available under license
- 1.196 kotlin-scripting-jvm 1.7.20**
 - 1.196.1 Available under license
- 1.197 google-go-cmp v0.5.6**
 - 1.197.1 Available under license
- 1.198 junit-platform-junit-platform-commons 1.8.2**
 - 1.198.1 Available under license
- 1.199 jackson-bom 2.14.0**
 - 1.199.1 Available under license
- 1.200 apache-log4j-slf4j-binding 2.17.1**
 - 1.200.1 Available under license
- 1.201 swift-poet 1.0.0**
 - 1.201.1 Available under license
- 1.202 error_prone_annotations 2.4.0**
 - 1.202.1 Available under license
- 1.203 jetbrains-kotlin-kotlin-stdlib-jdk7 1.4.10**
 - 1.203.1 Available under license
- 1.204 godoc-text v0.1.0**
 - 1.204.1 Available under license
- 1.205 io-grpc-grpc-protobuf 1.39.0**
 - 1.205.1 Available under license
- 1.206 jimfs-parent 1.1**
 - 1.206.1 Available under license
- 1.207 apiguardian-apiguardian-api 1.1.2**
 - 1.207.1 Available under license
- 1.208 go-tomb-tomb 20180513-snapshot-d5d1b582**
 - 1.208.1 Available under license

1.209 netty-codec-redis 4.1.85.Final

1.209.1 Available under license

1.210 okio 2.8.0

1.210.1 Available under license

1.211 guava-internalfuturefailureaccess-and-internalfutures 1.0.1

1.211.1 Available under license

1.212 mockito-inline 3.11.2

1.212.1 Available under license

1.213 go-spew v1.1.1

1.213.1 Available under license

1.214 checker-qual 3.12.0

1.214.1 Available under license

1.215 io-grpc-grpc-testing 1.39.0

1.215.1 Available under license

1.216 io-grpc-grpc-core 1.39.0

1.216.1 Available under license

1.217 cenk/backoff v4.1.0

1.217.1 Available under license

1.218 apache-log4j-api 2.19.0

1.218.1 Available under license

1.219 io-grpc-grpc-stub 1.39.0

1.219.1 Available under license

1.220 aws-glue-schema-registry-common 1.1.9

1.220.1 Available under license

1.221 junit-jupiter-junit-jupiter-engine 5.8.2

1.221.1 Available under license

1.222 pgv-java-parent 0.6.1

1.222.1 Available under license

1.223 netty-project 4.1.85.Final

1.223.1 Available under license

1.224 testcontainers-junit-jupiter-extension 1.17.3

1.224.1 Available under license

1.225 lumberjack v2.0.0

1.225.1 Available under license

1.226 pmezard-go-difflib 1.0.0

1.226.1 Available under license

1.227 golang-protobuf v1.5.2

1.227.1 Available under license

1.228 objenesis 3.2

- 1.228.1 Available under license
- 1.229 io-grpc-grpc-protobuf-lite 1.39.0**
 - 1.229.1 Available under license
- 1.230 grpc-go v1.40.0**
 - 1.230.1 Available under license
- 1.231 yaml-for-go 20200512-snapshot-9f266ea9**
 - 1.231.1 Available under license
- 1.232 protoc-gen-validate v0.6.1**
 - 1.232.1 Available under license
- 1.233 prometheus-client v1.12.0**
 - 1.233.1 Available under license
- 1.234 matttproud-golang-protobuf-extensions v1.0.1**
 - 1.234.1 Available under license
- 1.235 mbknor-jackson-jsonschema 1.0.39**
 - 1.235.1 Available under license
- 1.236 junit-5-bill-of-materials 5.8.2**
 - 1.236.1 Available under license
- 1.237 hamcrest 1.3**
 - 1.237.1 Available under license
- 1.238 apache-commons-digester 1.8.1**
 - 1.238.1 Available under license

1.1 prometheus-procfs v0.7.3

1.1.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the

direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

procfs provides functions to retrieve system, kernel and process metrics from the pseudo-filesystem proc.

Copyright 2014-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

1.2 error_prone_annotations 2.7.1

1.2.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2014 The Error Prone Authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-
jar/com/google/errorprone/annotations/NoAllocation.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-
jar/com/google/errorprone/annotations/concurrent/LockMethod.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-
jar/com/google/errorprone/annotations/concurrent/UnlockMethod.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2017 The Error Prone Authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/OverridingMethodsMustInvokeSuper.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/DoNotCall.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/concurrent/GuardedBy.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/CheckReturnValue.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2016 The Error Prone Authors.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/FormatString.java

* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/CompatibleWith.java

* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/RestrictedApi.java

* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/MustBeClosed.java

* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/DoNotMock.java

* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/FormatMethod.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2015 The Error Prone Authors.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/Var.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/concurrent/LazyInit.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/IncompatibleModifiers.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/RequiredModifiers.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/Immutable.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/CompileTimeConstant.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/SuppressPackageLocation.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/ForOverride.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/CanIgnoreReturnValue.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2021 The Error Prone Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-

jar/com/google/errorprone/annotations/InlineMeValidationDisabled.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-
jar/com/google/errorprone/annotations/InlineMe.java

1.3 x-sys v0.1.0

1.3.1 Available under license :

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.4 jmes-path-query-library 1.12.349

1.4.1 Available under license :

No license file was found, but licenses were detected in source scan.

<name>Apache License, Version 2.0</name>

Found in path(s):

* /opt/cola/permits/1526005813_1673052070.512654/0/jmespath-java-1-12-349-jar/META-INF/maven/com.amazonaws/jmespath-java/pom.xml

1.5 amazon-kinesis-client-library-for-java

2.4.2

1.5.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<!--  
/*  
* Copyright 2019 Amazon.com, Inc. or its affiliates.  
* Licensed under the Apache License, Version 2.0 (the  
* "License"); you may not use this file except in compliance  
* with the License. You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/  
-->
```

Found in path(s):

```
*/opt/cola/permits/1526005946_1673042434.4260015/0/amazon-kinesis-client-2-4-2-jar/META-INF/maven/software.amazon.kinesis/amazon-kinesis-client/pom.xml
```

1.6 cespare-xxhash v1.1.0

1.6.1 Available under license :

Copyright (c) 2016 Caleb Spare

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.7 jacoco 0.8.7

1.7.1 Available under license :

Found license 'Eclipse Public License 1.0' in 'This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at'

Found license 'Eclipse Public License 1.0' in 'under the terms and conditions of the Eclipse Public License Version 2.0'

1.8 kotlin-script-runtime 1.7.20

1.8.1 Available under license :

Apache-2.0

1.9 apache-avro 1.11.1

1.9.1 Available under license :

Apache Avro

Copyright 2010-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

NUnit license acknowledgement:

| Portions Copyright 2002-2012 Charlie Poole or Copyright 2002-2004 James

| W. Newkirk, Michael C. Two, Alexei A. Vorontsov or Copyright 2000-2002

| Philip A. Craig

Based upon the representations of upstream licensors, it is understood that portions of the mapreduce API included in the Java implementation are licensed from various contributors under one or more contributor license agreements to Odiago, Inc. and were then contributed by Odiago to Apache Avro, which has now made them available under the Apache 2.0 license. The original file header text is:

| Licensed to Odiago, Inc. under one or more contributor license

| agreements. See the NOTICE file distributed with this work for
| additional information regarding copyright ownership. Odiago, Inc.
| licenses this file to you under the Apache License, Version 2.0
| (the "License"); you may not use this file except in compliance
| with the License. You may obtain a copy of the License at
|
| <https://www.apache.org/licenses/LICENSE-2.0>
|
| Unless required by applicable law or agreed to in writing, software
| distributed under the License is distributed on an "AS IS" BASIS,
| WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
| implied. See the License for the specific language governing
| permissions and limitations under the License.

The Odiago NOTICE at the time of the contribution:

| This product includes software developed by Odiago, Inc.
| (<https://www.wibidata.com>).

Apache Ivy includes the following in its NOTICE file:

| Apache Ivy
| Copyright 2007-2010 The Apache Software Foundation
|
| This product includes software developed by
| The Apache Software Foundation (<https://www.apache.org/>).
|
| Portions of Ivy were originally developed by
| Jayasoft SARL (<http://www.jayasoft.fr/>)
| and are licensed to the Apache Software Foundation under the
| "Software Grant License Agreement"
|
| SSH and SFTP support is provided by the JCraft JSch package,
| which is open source software, available under
| the terms of a BSD style license.
| The original software and related information is available
| at <http://www.jcraft.com/jsch/>.

Apache Log4Net includes the following in its NOTICE file:

| Apache log4net
| Copyright 2004-2015 The Apache Software Foundation
|
| This product includes software developed at
| The Apache Software Foundation (<https://www.apache.org/>).

csharp reflect serializers were contributed by Pitney Bowes Inc.

| Copyright 2019 Pitney Bowes Inc.
| Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
| You may obtain a copy of the License at <https://www.apache.org/licenses/LICENSE-2.0>.
| Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
| WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
| See the License for the specific language governing permissions and limitations under the License.

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed

with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

License for Guava classes included in this binary artifact:

Copyright: 2006-2015 The Guava Authors
License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)
Apache Avro
Copyright 2010-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

Based upon the representations of upstream licensors, it is understood that portions of the mapreduce API included in the Java implementation are licensed from various contributors under one or more contributor license agreements to Odiago, Inc. and were then contributed by Odiago to Apache Avro, which has now made them available under the Apache 2.0 license. The original file header text is:

```
| Licensed to Odiago, Inc. under one or more contributor license
| agreements. See the NOTICE file distributed with this work for
| additional information regarding copyright ownership. Odiago, Inc.
| licenses this file to you under the Apache License, Version 2.0
| (the "License"); you may not use this file except in compliance
| with the License. You may obtain a copy of the License at
|
| https://www.apache.org/licenses/LICENSE-2.0
|
| Unless required by applicable law or agreed to in writing, software
| distributed under the License is distributed on an "AS IS" BASIS,
```

| WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
| implied. See the License for the specific language governing
| permissions and limitations under the License.

The Odiago NOTICE at the time of the contribution:

| This product includes software developed by Odiago, Inc.
| (<https://www.wibidata.com>).

The documentation contains the default Apache Forrest skin.
Apache Forrest includes the following in its NOTICE file:

| Apache Forrest
| Copyright 2002-2007 The Apache Software Foundation.

| This product includes software developed at
| The Apache Software Foundation (<https://www.apache.org/>).

| See also the file LICENSE.txt

| -----
| The purpose of this NOTICE.txt file is to contain notices that are
| required by the copyright owner and their license. Some of the
| accompanying products have an attribution requirement, so see below.
| Other accompanying products do not require attribution, so are not listed.

| -----
| This product includes software developed by the OpenSymphony Group
| <http://www.opensymphony.com/>

| This product includes software developed for project Krysalis
| <http://www.krysalis.org/>

| This product includes software developed by Andy Clark.
| <https://people.apache.org/~andyc/neko/>

| This product includes software developed by the ExoLab Project
| <https://www.exolab.org/>

| This product includes software developed by TouchGraph LLC
| <https://www.touchgraph.com/>

| This product includes software developed by Marc De Scheemaeker
| <http://nanoxml.cyberelf.be/>

| This product includes software developed by the ANTLR project
| <https://wwwantlr.org/>

| This product includes software developed by Chaperon

| <http://chaperon.sourceforge.net/>

|

| This product includes software developed by Sal Mangano (included in the XSLT Cookbook published by O'Reilly)

| <https://www.oreilly.com/catalog/xsltckbk/>

|

| This product includes software developed by The Werken Company.

| <http://jaxen.werken.com/>

|

| This product includes software developed by the jfor project

| <http://www.jfor.org/>

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

License for the m4 macros used by the C++ implementation:

Files:

* lang/c++/m4/m4_ax_boost_system.m4

Copyright (c) 2008 Thomas Porschberg <thomas@randspringer.de>

Copyright (c) 2008 Michael Tindal

Copyright (c) 2008 Daniel Casimiro <dan.casimiro@gmail.com>

* lang/c++/m4/m4_ax_boost_asio.m4

Copyright (c) 2008 Thomas Porschberg <thomas@randspringer.de>

Copyright (c) 2008 Pete Greenwell <pete@mu.org>

* lang/c++/m4/m4_ax_boost_filesystem.m4

Copyright (c) 2009 Thomas Porschberg <thomas@randspringer.de>

Copyright (c) 2009 Michael Tindal

Copyright (c) 2009 Roman Rybalko <libtorrent@romanr.info>

* lang/c++/m4/m4_ax_boost_thread.m4

Copyright (c) 2009 Thomas Porschberg <thomas@randspringer.de>

Copyright (c) 2009 Michael Tindal

* lang/c++/m4/m4_ax_boost_regex.m4

Copyright (c) 2008 Thomas Porschberg <thomas@randspringer.de>

Copyright (c) 2008 Michael Tindal

* lang/c++/m4/m4_ax_boost_base.m4

Copyright (c) 2008 Thomas Porschberg <thomas@randspringer.de>

License text:

| Copying and distribution of this file, with or without modification, are
| permitted in any medium without royalty provided the copyright notice
| and this notice are preserved. This file is offered as-is, without any
| warranty.

License for the AVRO_BOOT_NO_TRAIT code in the C++ implementation:
File: lang/c++/api/Boost.hh

| Boost Software License - Version 1.0 - August 17th, 2003

|
| Permission is hereby granted, free of charge, to any person or organization
| obtaining a copy of the software and accompanying documentation covered by
| this license (the "Software") to use, reproduce, display, distribute,
| execute, and transmit the Software, and to prepare derivative works of the
| Software, and to permit third-parties to whom the Software is furnished to
| do so, all subject to the following:

|
| The copyright notices in the Software and this entire statement, including
| the above license grant, this restriction and the following disclaimer,
| must be included in all copies of the Software, in whole or in part, and
| all derivative works of the Software, unless such copies or derivative
| works are solely in the form of machine-executable object code generated by
| a source language processor.

| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
| IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
| FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT
| SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE
| FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE,
| ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
| DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

License for portions of idl.jj in the Java compiler implementation:

Portions of idl.jj were modeled after the example Java 1.5 parser included with JavaCC. For those portions:

Copyright (c) 2006, Sun Microsystems, Inc.

All rights reserved.

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are met:

| * Redistributions of source code must retain the above copyright notice,
| this list of conditions and the following disclaimer.

| * Redistributions in binary form must reproduce the above copyright
| notice, this list of conditions and the following disclaimer in the
| documentation and/or other materials provided with the distribution.

| * Neither the name of the Sun Microsystems, Inc. nor the names of its
| contributors may be used to endorse or promote products derived from
| this software without specific prior written permission.

| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
| AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
| IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
| ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
| LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
| CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
| SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
| INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
| CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
| ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
| THE POSSIBILITY OF SUCH DAMAGE.

License for Jackson, included in this binary artifact:

Copyright: 2007-2015 Tatu Saloranta and other contributors

Home page: <http://jackson.codehaus.org/>

License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

License for SLF4J, include in this binary artifact:

Copyright (c) 2004-2013 QOS.ch

All rights reserved.

Home page: <https://www.slf4j.org/>

License: <https://slf4j.org/license.html> (MIT license)

SLF4J license text (MIT):

| Permission is hereby granted, free of charge, to any person obtaining
| a copy of this software and associated documentation files (the
| "Software"), to deal in the Software without restriction, including

| without limitation the rights to use, copy, modify, merge, publish,
| distribute, sublicense, and/or sell copies of the Software, and to
| permit persons to whom the Software is furnished to do so, subject to
| the following conditions:

|
| The above copyright notice and this permission notice shall be
| included in all copies or substantial portions of the Software.

|
| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
| EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
| MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
| NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
| LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
| OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
| WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

License for snappy-java, included in this binary artifact:

Copyright: 2011 Taro L. Saito and other contributors
Home page: <http://www.xerial.org/>
License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

License for Apache Commons Compress, included in this binary artifact:

Copyright: 2004-2015 The Apache Software Foundation
Home page: <https://commons.apache.org/compress/>
License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

Commons Compress includes files derived from the LZMA SDK, version 9.20 (C/ and
CPP/7zip/), in the package org.apache.commons.compress.archivers.sevenz:

| LZMA SDK is placed in the public domain. (<https://www.7-zip.org/sdk.html>)

License for xz compression, included in this binary artifact:

Home page: <https://tukaani.org/xz/java.html>

| This Java implementation of XZ has been put into the public domain, thus you
| can do whatever you want with it. All the files in the package have been
| written by Lasse Collin, but some files are heavily based on public domain code
| written by Igor Pavlov.

License for Apache Commons Lang, included in this binary artifact:

Copyright: 2002-2014 The Apache Software Foundation
Home page: <https://commons.apache.org/lang/>
License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

Commons Lang includes software from the Spring Framework, under the Apache License 2.0:

* `StringUtils.containsWhitespace()`

License for Apache Velocity, included in this binary artifact:

Copyright: 2000-2015 The Apache Software Foundation
Home page: <https://velocity.apache.org/>
License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

License for Apache Commons Collections, included in this binary artifact:

Copyright: 2001-2015 The Apache Software Foundation
Home page: <https://commons.apache.org/proper/commons-collections/>
License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

License for Jetty, included in this binary artifact:

Copyright: 1995-2015 Mort Bay Consulting Pty Ltd.
Home page: <https://eclipse.org/jetty/licenses.php>
License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

License for Netty, included in this binary artifact:

Copyright: 2011-2013 The Netty Project
Home page: <https://netty.io/>
License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

Netty contains the following code (copied from its NOTICE file with licenses added inline):

| This product contains the extensions to Java Collections Framework which has
| been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

|

| * LICENSE:

|| The person or persons who have associated work with this document (the
|| "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of
|| his knowledge, the work of authorship identified is in the public domain of
|| the country from which the work is published, or (b) hereby dedicates whatever
|| copyright the dedicators holds in the work of authorship identified below (the
|| "Work") to the public domain. A certifier, moreover, dedicates any copyright

|| interest he may have in the associated work, and for these purposes, is
|| described as a "dedicator" below.

||

|| A certifier has taken reasonable steps to verify the copyright status of this
|| work. Certifier recognizes that his good faith efforts may not shield him from
|| liability if in fact the work certified is not in the public domain.

||

|| Dedicator makes this dedication for the benefit of the public at large and to
|| the detriment of the Dedicator's heirs and successors. Dedicator intends this
|| dedication to be an overt act of relinquishment in perpetuate of all present
|| and future rights under copyright law, whether vested or contingent, in the
|| Work. Dedicator understands that such relinquishment of all rights includes
|| the relinquishment of all rights to enforce (by lawsuit or otherwise) those
|| copyrights in the Work.

||

|| Dedicator recognizes that, once placed in the public domain, the Work may be
|| freely reproduced, distributed, transmitted, used, modified, built upon, or
|| otherwise exploited by anyone for any purpose, commercial or non-commercial,
|| and in any way, including by methods that have not yet been invented or
|| conceived.

| * HOMEPAGE:

| * <http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/>

| * <http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jboss-cache/experimental/jsr166/>

|

| This product contains a modified version of Robert Harder's Public Domain
| Base64 Encoder and Decoder, which can be obtained at:

|

| * LICENSE: public domain (see JSR-166 license above)

| * HOMEPAGE:

| * <http://iharder.sourceforge.net/current/java/base64/>

|

| This product contains a modified version of 'JZlib', a re-implementation of
| zlib in pure Java, which can be obtained at:

|

| * LICENSE:

|| Copyright (c) 2000,2001,2002,2003,2004 ymnk, JCraft,Inc. All rights reserved.

||

|| Redistribution and use in source and binary forms, with or without
|| modification, are permitted provided that the following conditions are met:

||

|| 1. Redistributions of source code must retain the above copyright notice,
|| this list of conditions and the following disclaimer.

||

|| 2. Redistributions in binary form must reproduce the above copyright
|| notice, this list of conditions and the following disclaimer in
|| the documentation and/or other materials provided with the distribution.

||

|| 3. The names of the authors may not be used to endorse or promote products

|| derived from this software without specific prior written permission.

||

|| THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES,
|| INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
|| FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT,
|| INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT,
|| INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
|| LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA,
|| OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
|| LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
|| NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
|| EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

| * HOMEPAGE:

| * <http://www.jcraft.com/jzlib/>

License for the javax.servlet API, included in this binary artifact:

Copyright (c) 2003-2004 The Apache Software Foundation

License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

Source: <http://grepcode.com/project/repo1.maven.org/maven2/javax.servlet/servlet-api/>

License for Apache Commons Codec, included in this binary artifact:

Copyright: 2002-2015 The Apache Software Foundation

Home page: <https://commons.apache.org/proper/commons-codec/>

License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

License for Apache Commons CLI, included in this binary artifact:

Copyright: 2001-2015 The Apache Software Foundation

Home page: <https://commons.apache.org/proper/commons-cli/>

License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

License for Apache Commons Logging, included in this binary artifact:

Copyright: 2002-2014 The Apache Software Foundation

Home page: <https://commons.apache.org/proper/commons-logging/>

License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

License for Apache Commons HttpClient, included in this binary artifact:

Copyright: 1999-2005 The Apache Software Foundation

License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

License for Apache Hadoop, included in this binary artifact:

Copyright: 2001-2015 The Apache Software Foundation
Home page: <https://commons.apache.org/proper/commons-collections/>
License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

Hadoop contains the following code (from its LICENSE file):

The org.apache.hadoop.util.bloom.* classes:
| Copyright (c) 2005, European Commission project OneLab under contract
| 034819 (<http://www.one-lab.org>)
| All rights reserved.
| Redistribution and use in source and binary forms, with or
| without modification, are permitted provided that the following
| conditions are met:
| - Redistributions of source code must retain the above copyright
| notice, this list of conditions and the following disclaimer.
| - Redistributions in binary form must reproduce the above copyright
| notice, this list of conditions and the following disclaimer in
| the documentation and/or other materials provided with the distribution.
| - Neither the name of the University Catholique de Louvain - UCL
| nor the names of its contributors may be used to endorse or
| promote products derived from this software without specific prior
| written permission.
|
| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
| "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
| LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
| FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
| COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
| INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
| BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
| LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
| CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
| LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
| ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
| POSSIBILITY OF SUCH DAMAGE.

License for Google Guava, included in this binary artifact:

Copyright: 2007-2015 The Guava Authors
Home page: <https://github.com/google/guava>
License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for Apache Commons Math, included in this binary artifact:

Copyright: 2001-2015 The Apache Software Foundation

Home page: <https://commons.apache.org/proper/commons-math/>

License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

Commons Math includes other works under licenses compatible with the Apache Software License:

| APACHE COMMONS MATH DERIVATIVE WORKS:

|
| The Apache commons-math library includes a number of subcomponents
| whose implementation is derived from original sources written
| in C or Fortran. License terms of the original sources
| are reproduced below.

| =====

| For the lmdcr, lmpar and qrsolv Fortran routine from minpack and translated in
| the LevenbergMarquardtOptimizer class in package
| org.apache.commons.math3.optimization.general
| Original source copyright and license statement:

| Minpack Copyright Notice (1999) University of Chicago. All rights reserved

| Redistribution and use in source and binary forms, with or
| without modification, are permitted provided that the
| following conditions are met:

| 1. Redistributions of source code must retain the above
| copyright notice, this list of conditions and the following
| disclaimer.

| 2. Redistributions in binary form must reproduce the above
| copyright notice, this list of conditions and the following
| disclaimer in the documentation and/or other materials
| provided with the distribution.

| 3. The end-user documentation included with the
| redistribution, if any, must include the following
| acknowledgment:

| "This product includes software developed by the
| University of Chicago, as Operator of Argonne National
| Laboratory.

| Alternately, this acknowledgment may appear in the software
| itself, if and wherever such third-party acknowledgments
| normally appear.

| 4. WARRANTY DISCLAIMER. THE SOFTWARE IS SUPPLIED "AS IS"
| WITHOUT WARRANTY OF ANY KIND. THE COPYRIGHT HOLDER, THE
| UNITED STATES, THE UNITED STATES DEPARTMENT OF ENERGY, AND
| THEIR EMPLOYEES: (1) DISCLAIM ANY WARRANTIES, EXPRESS OR
| IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES
| OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE
| OR NON-INFRINGEMENT, (2) DO NOT ASSUME ANY LEGAL LIABILITY
| OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS, OR
| USEFULNESS OF THE SOFTWARE, (3) DO NOT REPRESENT THAT USE OF
| THE SOFTWARE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS, (4)
| DO NOT WARRANT THAT THE SOFTWARE WILL FUNCTION
| UNINTERRUPTED, THAT IT IS ERROR-FREE OR THAT ANY ERRORS WILL
| BE CORRECTED.

| 5. LIMITATION OF LIABILITY. IN NO EVENT WILL THE COPYRIGHT
| HOLDER, THE UNITED STATES, THE UNITED STATES DEPARTMENT OF
| ENERGY, OR THEIR EMPLOYEES: BE LIABLE FOR ANY INDIRECT,
| INCIDENTAL, CONSEQUENTIAL, SPECIAL OR PUNITIVE DAMAGES OF
| ANY KIND OR NATURE, INCLUDING BUT NOT LIMITED TO LOSS OF
| PROFITS OR LOSS OF DATA, FOR ANY REASON WHATSOEVER, WHETHER
| SUCH LIABILITY IS ASSERTED ON THE BASIS OF CONTRACT, TORT
| (INCLUDING NEGLIGENCE OR STRICT LIABILITY), OR OTHERWISE,
| EVEN IF ANY OF SAID PARTIES HAS BEEN WARNED OF THE
| POSSIBILITY OF SUCH LOSS OR DAMAGES.

=====

| Copyright and license statement for the odex Fortran routine developed by
| E. Hairer and G. Wanner and translated in GraggBulirschStoerIntegrator class
| in package org.apache.commons.math3.ode.nonstiff:

| Copyright (c) 2004, Ernst Hairer

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are
| met:

| - Redistributions of source code must retain the above copyright
| notice, this list of conditions and the following disclaimer.

| - Redistributions in binary form must reproduce the above copyright
| notice, this list of conditions and the following disclaimer in the
| documentation and/or other materials provided with the distribution.

| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
| IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
| TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
| PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR

| CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
| EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
| PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
| PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
| LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
| NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
| SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

| =====

|
| Copyright and license statement for the original lapack fortran routines
| translated in EigenDecompositionImpl class in package
| org.apache.commons.math3.linear:

| Copyright (c) 1992-2008 The University of Tennessee. All rights reserved.

| \$COPYRIGHT\$

| Additional copyrights may follow

| \$HEADERS\$

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are
| met:

- | - Redistributions of source code must retain the above copyright
| notice, this list of conditions and the following disclaimer.
- | - Redistributions in binary form must reproduce the above copyright
| notice, this list of conditions and the following disclaimer listed
| in this license in the documentation and/or other materials
| provided with the distribution.
- | - Neither the name of the copyright holders nor the names of its
| contributors may be used to endorse or promote products derived from
| this software without specific prior written permission.

| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
| "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
| LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
| A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
| OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
| SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
| LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
| DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
| THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
| (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
| OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

| Copyright and license statement for the original Mersenne twister C
| routines translated in MersenneTwister class in package
| org.apache.commons.math3.random:

| Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura,
| All rights reserved.

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions
| are met:

- | 1. Redistributions of source code must retain the above copyright
| notice, this list of conditions and the following disclaimer.
- | 2. Redistributions in binary form must reproduce the above copyright
| notice, this list of conditions and the following disclaimer in the
| documentation and/or other materials provided with the distribution.
- | 3. The names of its contributors may not be used to endorse or promote
| products derived from this software without specific prior written
| permission.

| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
| "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
| LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
| A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
| CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
| EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
| PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
| PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
| LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
| NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
| SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

| The class "org.apache.commons.math3.exception.util.LocalizedFormatsTest" is
| an adapted version of "OrekitMessagesTest" test class for the Orekit library
| The "org.apache.commons.math3.analysis.interpolation.HermiteInterpolator"
| has been imported from the Orekit space flight dynamics library.

| Th Orekit library is described at:
| <https://www.orekit.org/forge/projects/orekit>
| The original files are distributed under the terms of the Apache 2 license
| which is: Copyright 2010 CS Communication & Systmes

License for XMLenc, included in this binary artifact:

Copyright 2003-2011, Ernst de Haan
All rights reserved.

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are met:

| 1. Redistributions of source code must retain the above copyright notice, this
| list of conditions and the following disclaimer.

| 2. Redistributions in binary form must reproduce the above copyright notice,
| this list of conditions and the following disclaimer in the documentation
| and/or other materials provided with the distribution.

| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
| AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
| IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
| DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
| FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
| DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
| SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
| CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
| OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
| OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License for Apache Commons IO, included in this binary artifact:

Copyright: 2002-2015 The Apache Software Foundation
Home page: <https://commons.apache.org/proper/io/>
License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for Apache Commons Net, included in this binary artifact:

Copyright: 2001-2015 The Apache Software Foundation
Home page: <https://commons.apache.org/proper/commons-net/>
License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for Apache Log4j, included in this binary artifact:

Copyright: 1999-2015 The Apache Software Foundation
Home page: <https://logging.apache.org/log4j/>
License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for Apache Commons Configuration, included in this binary artifact:

Copyright: 2001-2015 The Apache Software Foundation
Home page: <https://commons.apache.org/proper/commons-configuration/>
License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for Apache Commons Digester, included in this binary artifact:

Copyright: 2001-2015 The Apache Software Foundation
License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for Apache Commons Beanutils, included in this binary artifact:

Copyright: 2000-2015 The Apache Software Foundation
License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for Google Protocol Buffers, included in this binary artifact:

Copyright 2014, Google Inc. All rights reserved.

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are
| met:

| * Redistributions of source code must retain the above copyright
| notice, this list of conditions and the following disclaimer.

| * Redistributions in binary form must reproduce the above
| copyright notice, this list of conditions and the following disclaimer
| in the documentation and/or other materials provided with the
| distribution.

| * Neither the name of Google Inc. nor the names of its
| contributors may be used to endorse or promote products derived from
| this software without specific prior written permission.

| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
| "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
| LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
| A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
| OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
| SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
| LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
| DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
| THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
| (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

| OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

|

| Code generated by the Protocol Buffer compiler is owned by the owner
| of the input file used when generating it. This code is not
| standalone and requires a support library to be linked with it. This
| support library is itself covered by the above license.

License for Apache HttpClient, included in this binary artifact:

Copyright: 1999-2015 The Apache Software Foundation
License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

HttpClient contains the following data under the terms of the MPL:

| This project includes Public Suffix List copied from
| <https://publicsuffix.org/list/effective_tld_names.dat>
| licensed under the terms of the Mozilla Public License, v. 2.0
|
| Full license text: META-INF/mpl-2.0.text

License for Apache Directory, included in this binary artifact:

Copyright: 2003-2015 The Apache Software Foundation
License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

Apache Directory includes other works under licenses compatible with the
Apache Software License:

| -----

|

| The OpenLDAP Public License
| Version 2.8, 17 August 2003

|

| Redistribution and use of this software and associated documentation
| ("Software"), with or without modification, are permitted provided
| that the following conditions are met:

|

| 1. Redistributions in source form must retain copyright statements
| and notices,

|

| 2. Redistributions in binary form must reproduce applicable copyright
| statements and notices, this list of conditions, and the following
| disclaimer in the documentation and/or other materials provided
| with the distribution, and

|

| 3. Redistributions must contain a verbatim copy of this document.

|

| The OpenLDAP Foundation may revise this license from time to time.
| Each revision is distinguished by a version number. You may use
| this Software under terms of this license revision or under the
| terms of any subsequent revision of the license.

| THIS SOFTWARE IS PROVIDED BY THE OPENLDAP FOUNDATION AND ITS
| CONTRIBUTORS ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES,
| INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
| AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
| SHALL THE OPENLDAP FOUNDATION, ITS CONTRIBUTORS, OR THE AUTHOR(S)
| OR OWNER(S) OF THE SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT,
| INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
| BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
| LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
| CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
| LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
| ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
| POSSIBILITY OF SUCH DAMAGE.

| The names of the authors and copyright holders must not be used in
| advertising or otherwise to promote the sale, use or other dealing
| in this Software without specific, written prior permission. Title
| to copyright in this Software shall at all times remain with copyright
| holders.

| OpenLDAP is a registered trademark of the OpenLDAP Foundation.

| Copyright 1999-2003 The OpenLDAP Foundation, Redwood City,
| California, USA. All Rights Reserved. Permission to copy and
| distribute verbatim copies of this document is granted.

| -----

| Copyright (c) 2000 - 2011 The Legion Of The Bouncy Castle (<https://www.bouncycastle.org>)

| Permission is hereby granted, free of charge, to any person obtaining a
| copy of this software and associated documentation files (the "Software"),
| to deal in the Software without restriction, including without limitation
| the rights to use, copy, modify, merge, publish, distribute, sublicense,
| and/or sell copies of the Software, and to permit persons to whom the
| Software is furnished to do so, subject to the following conditions:

| The above copyright notice and this permission notice shall be included in
| all copies or substantial portions of the Software.

| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
| IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
| FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

| AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
| LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
| OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
| THE SOFTWARE.

|
|

=====
| slf4j 1.7.10 license:

| -----

| Copyright (c) 2004-2013 QOS.ch

| All rights reserved.

|

| Permission is hereby granted, free of charge, to any person obtaining
| a copy of this software and associated documentation files (the
| "Software"), to deal in the Software without restriction, including
| without limitation the rights to use, copy, modify, merge, publish,
| distribute, sublicense, and/or sell copies of the Software, and to
| permit persons to whom the Software is furnished to do so, subject to
| the following conditions:

|

| The above copyright notice and this permission notice shall be
| included in all copies or substantial portions of the Software.

|

| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
| EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
| MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
| NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
| LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
| OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
| WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

|

| =====

| For the AVL Set code : <http://bobah.net/d4d/source-code/misc/java-avl-tree>

| -----

| Copyright 2001-2014 Vladimir Lysyy

| Licensed under the Apache License, Version 2.0 (the "License");

| you may not use this source code except in compliance with the License.

| You may obtain a copy of the License at

|

| <https://www.apache.org/licenses/LICENSE-2.0>

|

| Unless required by applicable law or agreed to in writing, software
| distributed under the License is distributed on an "AS IS" BASIS,
| WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

| See the License for the specific language governing permissions and

| limitations under the License.

License for the JSR-305 annotations, included in this binary artifact:

Copyright: 2011-2015 Stephen Connolly, Greg Lucas
License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for Apache ZooKeeper, included in this binary artifact:

Copyright: 2009-2015 The Apache Software Foundation
License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for Jersey, included in this binary artifact:

Copyright (c) 2015 Oracle and/or its affiliates.
All rights reserved.
License: CDDL 1.1: META-INF/cddl-1.1.text
Source: <https://github.com/jersey/jersey-1.x-old>

License for LevelDB JNI, included in this binary artifact:

Copyright (c) 2011 FuseSource Corp. All rights reserved.

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are
| met:

| * Redistributions of source code must retain the above copyright
| notice, this list of conditions and the following disclaimer.
| * Redistributions in binary form must reproduce the above
| copyright notice, this list of conditions and the following disclaimer
| in the documentation and/or other materials provided with the
| distribution.
| * Neither the name of FuseSource Corp. nor the names of its
| contributors may be used to endorse or promote products derived from
| this software without specific prior written permission.

| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
| "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
| LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
| A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
| OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
| SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
| LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
| DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
| THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

| (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
| OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License for iq80 LevelDB Java API, included in this binary artifact:

Copyright 2011 Dain Sundstrom <dain@iq80.com>
Copyright 2011 FuseSource Corp. <http://fusesource.com>
License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for jquery and jquery-ui, included in this binary artifact:

License: The MIT License (MIT): <https://tldrlegal.com/license/mit-license>
Home page: <https://jquery.org/license/>

Copyright (c) <year> <copyright holders>

| Permission is hereby granted, free of charge, to any person obtaining a copy of
| this software and associated documentation files (the "Software"), to deal in
| the Software without restriction, including without limitation the rights to
| use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
| of the Software, and to permit persons to whom the Software is furnished to do
| so, subject to the following conditions:

|
| The above copyright notice and this permission notice shall be included in all
| copies or substantial portions of the Software.

|
| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
| IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
| FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
| AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
| LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
| OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
| SOFTWARE.

License for the javax.xml.bind jaxb API, included in this binary artifact:

Copyright (c) 2004-2010 Oracle and/or its affiliates.
All rights reserved.
License: CDDL 1.0: META-INF/cddl-1.0.text
Source: <http://www.grepcode.com/project/repo1.maven.org/maven2/javax.xml.bind/jaxb-api/>

License for the javax.xml.stream stax API, included in this binary artifact:

Copyright (c) 2004-2006 Oracle and/or its affiliates.

All rights reserved.

License: CDDL 1.0: META-INF/cddl-1.0.text

Source: <http://grepcode.com/project/repo1.maven.org/maven2/javax.xml.stream/stax-api/>

License for the javax.activation API, included in this binary artifact:

Copyright (c) 2004-2006 Oracle and/or its affiliates.

All rights reserved.

License: CDDL 1.0: META-INF/cddl-1.0.text

Source: <http://grepcode.com/project/repo1.maven.org/maven2/javax.activation/activation/>

License for the javax.ws.rs API, included in this binary artifact:

Copyright (c) 1996-2015, Oracle Corporation and/or its affiliates.

All rights reserved.

License: CDDL 1.1: META-INF/cddl-1.1.text

Source: <http://grepcode.com/project/repo1.maven.org/maven2/javax.ws.rs/javax.ws.rs-api/>

License for JOpt Simple, included in this binary artifact:

Copyright (c) 2004-2015 Paul R. Holser, Jr.

| Permission is hereby granted, free of charge, to any person obtaining
| a copy of this software and associated documentation files (the
| "Software"), to deal in the Software without restriction, including
| without limitation the rights to use, copy, modify, merge, publish,
| distribute, sublicense, and/or sell copies of the Software, and to
| permit persons to whom the Software is furnished to do so, subject to
| the following conditions:

|
| The above copyright notice and this permission notice shall be
| included in all copies or substantial portions of the Software.

|
| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
| EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
| MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
| NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
| LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
| OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
| WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

License for SLF4J API, included in this binary artifact:

Copyright (c) 2004-2013 QOS.ch

All rights reserved.

| Permission is hereby granted, free of charge, to any person obtaining
| a copy of this software and associated documentation files (the
| "Software"), to deal in the Software without restriction, including
| without limitation the rights to use, copy, modify, merge, publish,
| distribute, sublicense, and/or sell copies of the Software, and to
| permit persons to whom the Software is furnished to do so, subject to
| the following conditions:

|
| The above copyright notice and this permission notice shall be
| included in all copies or substantial portions of the Software.

| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
| EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
| MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
| NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
| LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
| OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
| WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

License for Guava classes included in this binary artifact:

Copyright: 2006-2015 The Guava Authors

License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing

the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

License for the Jansson C JSON parser used in the C implementation:

Copyright (c) 2009-2011 Petri Lehtinen <petri@digip.org>

Some files include an additional copyright notice:

* lang/c/jansson/src/pack_unpack.c

Copyright (c) 2011 Graeme Smecher <graeme.smecher@mail.mcgill.ca>

* lang/c/jansson/test/suites/api/test_unpack.c

Copyright (c) 2011 Graeme Smecher <graeme.smecher@mail.mcgill.ca>

* lang/c/jansson/src/memory.c

Copyright (c) 2011 Basile Starynkevitch <basile@starynkevitch.net>

| Permission is hereby granted, free of charge, to any person obtaining a copy
| of this software and associated documentation files (the "Software"), to deal
| in the Software without restriction, including without limitation the rights
| to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
| copies of the Software, and to permit persons to whom the Software is
| furnished to do so, subject to the following conditions:

|
| The above copyright notice and this permission notice shall be included in
| all copies or substantial portions of the Software.

|
| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
| IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
| FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
| AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
| LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
| OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
| THE SOFTWARE.

License for msinttypes.h and msstdint.h used in the C implementation:

Source from:

<https://code.google.com/p/msinttypes/downloads/detail?name=msinttypes-r26.zip>

Copyright (c) 2006-2008 Alexander Chemeris

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are met:

- |
- | 1. Redistributions of source code must retain the above copyright notice,
| this list of conditions and the following disclaimer.
 - | 2. Redistributions in binary form must reproduce the above copyright
| notice, this list of conditions and the following disclaimer in the
| documentation and/or other materials provided with the distribution.
 - | 3. The name of the author may be used to endorse or promote products
| derived from this software without specific prior written permission.
- |

| THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED
| WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
| MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
| EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
| SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
| PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;
| OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
| WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR

| OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
| ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License for st.c and st.h used in the C implementation:

| This is a public domain general purpose hash table package written by
| Peter Moore @ UCB.

License for Diredt API for Microsoft Visual Studio used in the C implementation:

Source from:

<http://www.softagalleria.net/download/diredt/diredt-1.11.zip>

Copyright (C) 2006 Toni Ronkko

| Permission is hereby granted, free of charge, to any person obtaining
| a copy of this software and associated documentation files (the
| ``Software"), to deal in the Software without restriction, including
| without limitation the rights to use, copy, modify, merge, publish,
| distribute, sublicense, and/or sell copies of the Software, and to
| permit persons to whom the Software is furnished to do so, subject to
| the following conditions:

| The above copyright notice and this permission notice shall be included
| in all copies or substantial portions of the Software.

| THE SOFTWARE IS PROVIDED ``AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
| OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
| MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
| IN NO EVENT SHALL TONI RONKKO BE LIABLE FOR ANY CLAIM, DAMAGES OR
| OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
| ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
| OTHER DEALINGS IN THE SOFTWARE.

License for ivy-2.2.0.jar used in the python implementation:

Apache License version 2.0 (see above)

License for pyAntTasks-1.3.jar used in the python implementation:

Apache License version 2.0 (see above)

License for NUnit binary included with the C# implementation:

File: nunit.framework.dll

| NUnit License

|

| Copyright 2002-2015 Charlie Poole

| Copyright 2002-2004 James W. Newkirk, Michael C. Two, Alexei A. Vorontsov

| Copyright 2000-2002 Philip A. Craig

|

| This software is provided 'as-is', without any express or implied warranty. In
| no event will the authors be held liable for any damages arising from the use
| of this software.

|

| Permission is granted to anyone to use this software for any purpose, including
| commercial applications, and to alter it and redistribute it freely, subject to
| the following restrictions:

|

| The origin of this software must not be misrepresented; you must not claim that
| you wrote the original software. If you use this software in a product, an
| acknowledgment (see the following) in the product documentation is required.

|

| Portions Copyright 2002-2012 Charlie Poole or Copyright 2002-2004 James W.
| Newkirk, Michael C. Two, Alexei A. Vorontsov or Copyright 2000-2002 Philip A.
| Craig

|

| Altered source versions must be plainly marked as such, and must not be
| misrepresented as being the original software.

|

| This notice may not be removed or altered from any source distribution.

| License Note

|

| This license is based on the open source zlib/libpng license. The idea was to
| keep the license as simple as possible to encourage use of NUnit in free and
| commercial applications and libraries, but to keep the source code together and
| to give credit to the NUnit contributors for their efforts. While this license
| allows shipping NUnit in source and binary form, if shipping a NUnit variant is
| the sole purpose of your product, please let us know.

License for the Json.NET binary included with the C# implementation:

File: Newtonsoft.Json.dll

Copyright (c) 2007 James Newton-King

| Permission is hereby granted, free of charge, to any person obtaining
| a copy of this software and associated documentation files (the
| "Software"), to deal in the Software without restriction, including
| without limitation the rights to use, copy, modify, merge, publish,
| distribute, sublicense, and/or sell copies of the Software, and to
| permit persons to whom the Software is furnished to do so, subject to

| the following conditions:

|

| The above copyright notice and this permission notice shall be
| included in all copies or substantial portions of the Software.

|

| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
| EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
| MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
| NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
| LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
| OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
| WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

License for the Castle Core binary included with the C# implementation:

File: Castle.Core.dll

Copyright (c) 2004-2015 Castle Project

License: Apache License version 2.0 (see above)

URL: <https://opensource.org/licenses/Apache-2.0>

License for the log4net binary included with the C# implementation:

File: log4net.dll

Copyright 2004-2015 The Apache Software Foundation.

License: Apache License version 2.0 (see above)

License for the m4 macros used by the C++ implementation:

Files:

* lang/c++/m4/m4_ax_boost_system.m4

Copyright (c) 2008 Thomas Porschberg <thomas@randspringer.de>

Copyright (c) 2008 Michael Tindal

Copyright (c) 2008 Daniel Casimiro <dan.casimiro@gmail.com>

* lang/c++/m4/m4_ax_boost_asio.m4

Copyright (c) 2008 Thomas Porschberg <thomas@randspringer.de>

Copyright (c) 2008 Pete Greenwell <pete@mu.org>

* lang/c++/m4/m4_ax_boost_filesystem.m4

Copyright (c) 2009 Thomas Porschberg <thomas@randspringer.de>

Copyright (c) 2009 Michael Tindal

Copyright (c) 2009 Roman Rybalko <libtorrent@romanr.info>

* lang/c++/m4/m4_ax_boost_thread.m4

Copyright (c) 2009 Thomas Porschberg <thomas@randspringer.de>

Copyright (c) 2009 Michael Tindal

* lang/c++/m4/m4_ax_boost_regex.m4
Copyright (c) 2008 Thomas Porschberg <thomas@randspringer.de>
Copyright (c) 2008 Michael Tindal
* lang/c++/m4/m4_ax_boost_base.m4
Copyright (c) 2008 Thomas Porschberg <thomas@randspringer.de>

License text:

| Copying and distribution of this file, with or without modification, are
| permitted in any medium without royalty provided the copyright notice
| and this notice are preserved. This file is offered as-is, without any
| warranty.

License for the AVRO_BOOT_NO_TRAIT code in the C++ implementation:
File: lang/c++/api/Boost.hh

| Boost Software License - Version 1.0 - August 17th, 2003

|
| Permission is hereby granted, free of charge, to any person or organization
| obtaining a copy of the software and accompanying documentation covered by
| this license (the "Software") to use, reproduce, display, distribute,
| execute, and transmit the Software, and to prepare derivative works of the
| Software, and to permit third-parties to whom the Software is furnished to
| do so, all subject to the following:

|
| The copyright notices in the Software and this entire statement, including
| the above license grant, this restriction and the following disclaimer,
| must be included in all copies of the Software, in whole or in part, and
| all derivative works of the Software, unless such copies or derivative
| works are solely in the form of machine-executable object code generated by
| a source language processor.

|
| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
| IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
| FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT
| SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE
| FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE,
| ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
| DEALINGS IN THE SOFTWARE.

License for jquery.tipsy.js, tipsy.js, and tipsy.css used by the Java IPC implementation:

Copyright (c) 2008 Jason Frame (jason@onehackoranother.com)

| Permission is hereby granted, free of charge, to any person obtaining a copy
| of this software and associated documentation files (the "Software"), to deal
| in the Software without restriction, including without limitation the rights

| to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
| copies of the Software, and to permit persons to whom the Software is
| furnished to do so, subject to the following conditions:

|
| The above copyright notice and this permission notice shall be included in
| all copies or substantial portions of the Software.

|
| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
| IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
| FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
| AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
| LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
| OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
| THE SOFTWARE.

License for protovis-r3.2.js used by the Java IPC implementation:

Copyright (c) 2010, Stanford Visualization Group
All rights reserved.

| Redistribution and use in source and binary forms, with or without modification,
| are permitted provided that the following conditions are met:

| * Redistributions of source code must retain the above copyright notice,
| this list of conditions and the following disclaimer.

| * Redistributions in binary form must reproduce the above copyright notice,
| this list of conditions and the following disclaimer in the documentation
| and/or other materials provided with the distribution.

| * Neither the name of Stanford University nor the names of its contributors
| may be used to endorse or promote products derived from this software
| without specific prior written permission.

| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
| ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
| WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
| DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR
| ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
| (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
| LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
| ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
| (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
| SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License for g.Raphael 0.4.1 used by the Java IPC implementation:

Copyright (c) 2009 Dmitry Baranovskiy (<http://g.raphaeljs.com>)
Licensed under the MIT (<https://www.opensource.org/licenses/mit-license.php>) license.

License for jQuery v1.4.2 used by the Java IPC implementation:

Copyright 2010, John Resig
Dual licensed under the MIT or GPL Version 2 licenses.
<https://jquery.org/license>

jQuery includes Sizzle.js
<https://sizzlejs.com/>
Copyright 2010, The Dojo Foundation
Released under the MIT, BSD, and GPL Licenses.

Both are included under the terms of the MIT license:

| Permission is hereby granted, free of charge, to any person obtaining a copy
| of this software and associated documentation files (the "Software"), to deal
| in the Software without restriction, including without limitation the rights
| to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
| copies of the Software, and to permit persons to whom the Software is
| furnished to do so, subject to the following conditions:

|
| The above copyright notice and this permission notice shall be included in
| all copies or substantial portions of the Software.

|
| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
| IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
| FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
| AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
| LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
| OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
| THE SOFTWARE.

License for portions of idl.jj in the Java compiler implementation:

Portions of idl.jj were modeled after the example Java 1.5
parser included with JavaCC. For those portions:

Copyright (c) 2006, Sun Microsystems, Inc.
All rights reserved.

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are met:

|

| * Redistributions of source code must retain the above copyright notice,
| this list of conditions and the following disclaimer.
| * Redistributions in binary form must reproduce the above copyright
| notice, this list of conditions and the following disclaimer in the
| documentation and/or other materials provided with the distribution.
| * Neither the name of the Sun Microsystems, Inc. nor the names of its
| contributors may be used to endorse or promote products derived from
| this software without specific prior written permission.

| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
| AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
| IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
| ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
| LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
| CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
| SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
| INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
| CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
| ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
| THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,

defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Mozilla Public License
Version 2.0

1. Definitions

1.1. Contributor

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. Contributor Version

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. Contribution

means Covered Software of a particular Contributor.

1.4. Covered Software

means Source Code Form to which the initial Contributor has attached the notice

in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses

means

a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. Executable Form

means any form of the work other than Source Code Form.

1.7. Larger Work

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. License

means this document.

1.9. Licensable

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. Modifications

means any of the following:

a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

b. any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. Secondary License

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. Source Code Form

means the form of the work preferred for making modifications.

1.14. You (or Your)

means an individual or a legal entity exercising rights under this License. For legal entities, You includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of

goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <https://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible With Secondary Licenses, as defined by the Mozilla Public License, v. 2.0.

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed

as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

License for the Json.NET binary included with the C# implementation:

File: Newtonsoft.Json.dll

Copyright (c) 2007 James Newton-King

| Permission is hereby granted, free of charge, to any person obtaining
| a copy of this software and associated documentation files (the
| "Software"), to deal in the Software without restriction, including
| without limitation the rights to use, copy, modify, merge, publish,
| distribute, sublicense, and/or sell copies of the Software, and to
| permit persons to whom the Software is furnished to do so, subject to
| the following conditions:

|

| The above copyright notice and this permission notice shall be
| included in all copies or substantial portions of the Software.

|
| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
| EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
| MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
| NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
| LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
| OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
| WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

License for the Castle Core binary included with the C# implementation:

File: Castle.Core.dll

Copyright (c) 2004-2015 Castle Project

License: Apache License version 2.0 (see above)

URL: <https://opensource.org/licenses/Apache-2.0>

License for the log4net binary included with the C# implementation:

File: log4net.dll

Copyright 2004-2015 The Apache Software Foundation.

License: Apache License version 2.0 (see above)

Apache Avro

Copyright 2010 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

Apache Avro

Copyright 2010-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

License for the AVRO_BOOT_NO_TRAIT code in the C++ implementation:

File: lang/c++/api/Boost.hh

| Boost Software License - Version 1.0 - August 17th, 2003

|
| Permission is hereby granted, free of charge, to any person or organization
| obtaining a copy of the software and accompanying documentation covered by
| this license (the "Software") to use, reproduce, display, distribute,
| execute, and transmit the Software, and to prepare derivative works of the
| Software, and to permit third-parties to whom the Software is furnished to
| do so, all subject to the following:

|
| The copyright notices in the Software and this entire statement, including
| the above license grant, this restriction and the following disclaimer,
| must be included in all copies of the Software, in whole or in part, and
| all derivative works of the Software, unless such copies or derivative
| works are solely in the form of machine-executable object code generated by
| a source language processor.

|
| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
| IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
| FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT
| SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE
| FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE,
| ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
| DEALINGS IN THE SOFTWARE.

License for jQuery v1.7.1 used in the C# documentation

Copyright 2010-2011, John Resig

Dual licensed under the MIT or GPL Version 2 licenses.

<https://jquery.org/license>

jQuery includes Sizzle.js

<https://sizzlejs.com/>

Copyright 2010-2011, The Dojo Foundation

Released under the MIT, BSD, and GPL Licenses.

Both are included under the terms of the MIT license:

| Permission is hereby granted, free of charge, to any person obtaining a copy
| of this software and associated documentation files (the "Software"), to deal
| in the Software without restriction, including without limitation the rights
| to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
| copies of the Software, and to permit persons to whom the Software is
| furnished to do so, subject to the following conditions:

|
| The above copyright notice and this permission notice shall be included in
| all copies or substantial portions of the Software.

|
| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
| IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
| FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
| AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
| LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
| OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
| THE SOFTWARE.

License for portions of idl.jj in the Java compiler implementation:

Portions of idl.jj were modeled after the example Java 1.5
parser included with JavaCC. For those portions:

Copyright (c) 2006, Sun Microsystems, Inc.
All rights reserved.

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are met:

- |
- | * Redistributions of source code must retain the above copyright notice,
| this list of conditions and the following disclaimer.
 - | * Redistributions in binary form must reproduce the above copyright
| notice, this list of conditions and the following disclaimer in the
| documentation and/or other materials provided with the distribution.
 - | * Neither the name of the Sun Microsystems, Inc. nor the names of its
| contributors may be used to endorse or promote products derived from
| this software without specific prior written permission.
- |

| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
| AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
| IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
| ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
| LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
| CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
| SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
| INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
| CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
| ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
| THE POSSIBILITY OF SUCH DAMAGE.

License for prototype.js included in the Avro documentation:

Prototype JavaScript framework, version 1.4.0_pre4
(c) 2005 Sam Stephenson <sam@conio.net>

| Prototype is freely distributable under the terms of an MIT-style license.
|
| For details, see the Prototype web site: <http://prototype.conio.net/>

For a copy of the MIT license text, see above.

License for Apache Forrest (skin), included in the Avro documentation:

Copyright: 2009-2015 The Apache Software Foundation
License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for Doxygen-generated documentation for the C++ and C# implementations:

Copyright 1997-2015 by Dimitri van Heesch.

| Doxygen license

|
| Permission to use, copy, modify, and distribute this software and its
| documentation under the terms of the GNU General Public License is hereby
| granted. No representations are made about the suitability of this software for
| any purpose. It is provided "as is" without express or implied warranty. See
| the GNU General Public License for more details.

|
| Documents produced by doxygen are derivative works derived from the input
| used in their production; they are not affected by this license.

Apache Avro

Copyright 2010-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

Apache Avro

Copyright 2011-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

title: "License"

linkTitle: "License"

weight: 3

manualLink: <https://www.apache.org/licenses/>

<!--

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

-->

Apache Avro project is licensed under [Apache Software License 2.0](<https://www.apache.org/licenses/LICENSE-2.0>)

Apache Avro

Copyright 2010-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

Based upon the representations of upstream licensors, it is understood that portions of the mapreduce API included in the Java implementation are licensed from various contributors under one or more contributor license agreements to Odiago, Inc. and were then contributed by Odiago to Apache Avro, which has now made them available under the Apache 2.0 license. The original file header text is:

| Licensed to Odiago, Inc. under one or more contributor license
| agreements. See the NOTICE file distributed with this work for
| additional information regarding copyright ownership. Odiago, Inc.
| licenses this file to you under the Apache License, Version 2.0
| (the "License"); you may not use this file except in compliance
| with the License. You may obtain a copy of the License at

| <https://www.apache.org/licenses/LICENSE-2.0>

| Unless required by applicable law or agreed to in writing, software
| distributed under the License is distributed on an "AS IS" BASIS,
| WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
| implied. See the License for the specific language governing

| permissions and limitations under the License.

The Odiago NOTICE at the time of the contribution:

| This product includes software developed by Odiago, Inc.
| (<https://www.wibidata.com>).

Apache Commons compress includes the following in its NOTICE file:

| Apache Commons Compress
| Copyright 2002-2014 The Apache Software Foundation
|
| This product includes software developed at
| The Apache Software Foundation (<https://www.apache.org/>).
|
| The files in the package org.apache.commons.compress.archivers.sevenz
| were derived from the LZMA SDK, version 9.20 (C/ and CPP/7zip/),
| which has been placed in the public domain:
|
| "LZMA SDK is placed in the public domain." (<https://www.7-zip.org/sdk.html>)

Apache Commons codec includes the following in its NOTICE file:

| Apache Commons Codec
| Copyright 2002-2015 The Apache Software Foundation
|
| This product includes software developed at
| The Apache Software Foundation (<https://www.apache.org/>).
|
| The content of package org.apache.commons.codec.language.bm has been translated
| from the original php source code available at <https://stevemorse.org/phoneticinfo.htm>
| with permission from the original authors.
| Original source copyright:
| Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

Apache Commons lang includes the following in its NOTICE file:

| Apache Commons Lang
| Copyright 2001-2011 The Apache Software Foundation
|
| This product includes software developed by
| The Apache Software Foundation (<https://www.apache.org/>).

Apache Velocity includes the following in its NOTICE file:

| Apache Velocity
| Copyright (C) 2000-2007 The Apache Software Foundation
|

| This product includes software developed at
| The Apache Software Foundation (<https://www.apache.org/>).

Apache Commons collections includes the following in its NOTICE file:

| Apache Commons Collections
| Copyright 2001-2008 The Apache Software Foundation
|
| This product includes software developed by
| The Apache Software Foundation (<https://www.apache.org/>).

Apache Commons math includes the following in its NOTICE file:

| =====
|
| The BracketFinder (package org.apache.commons.math3.optimization.univariate)
| and PowellOptimizer (package org.apache.commons.math3.optimization.general)
| classes are based on the Python code in module "optimize.py" (version 0.5)
| developed by Travis E. Oliphant for the SciPy library (<https://www.scipy.org/>)
| Copyright 2003-2009 SciPy Developers.

| =====
|
| The LinearConstraint, LinearObjectiveFunction, LinearOptimizer,
| Relationship, SimplexSolver and SimplexTableau classes in package
| org.apache.commons.math3.optimization.linear include software developed by
| Benjamin McCann (<https://www.benmccann.com>) and distributed with
| the following copyright: Copyright 2009 Google Inc.

| =====
|
| This product includes software developed by the
| University of Chicago, as Operator of Argonne National
| Laboratory.
| The LevenbergMarquardtOptimizer class in package
| org.apache.commons.math3.optimization.general includes software
| translated from the lmdcr, lmpar and qrsolv Fortran routines
| from the Minpack package
| Minpack Copyright Notice (1999) University of Chicago. All rights reserved

| =====
|
| The GraggBulirschStoerIntegrator class in package
| org.apache.commons.math3.ode.nonstiff includes software translated
| from the odex Fortran routine developed by E. Hairer and G. Wanner.
| Original source copyright:
| Copyright (c) 2004, Ernst Hairer

| =====
|
| The EigenDecompositionImpl class in package
| org.apache.commons.math3.linear includes software translated

| from some LAPACK Fortran routines. Original source copyright:
| Copyright (c) 1992-2008 The University of Tennessee. All rights reserved.

=====
|
| The MersenneTwister class in package org.apache.commons.math3.random
| includes software translated from the 2002-01-26 version of
| the Mersenne-Twister generator written in C by Makoto Matsumoto and Takuji
| Nishimura. Original source copyright:
| Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura,
| All rights reserved

=====
|
| The LocalizedFormatsTest class in the unit tests is an adapted version of
| the OrekitMessagesTest class from the orekit library distributed under the
| terms of the Apache 2 licence. Original source copyright:
| Copyright 2010 CS Systmes d'Information

=====
|
| The HermiteInterpolator class and its corresponding test have been imported from
| the orekit library distributed under the terms of the Apache 2 licence. Original
| source copyright:
| Copyright 2010-2012 CS Systmes d'Information

=====
|
| The creation of the package "o.a.c.m.analysis.integration.gauss" was inspired
| by an original code donated by Sbastien Brisard.

=====
|
| The complete text of licenses and disclaimers associated with the the original
| sources enumerated above at the time of code translation are in the LICENSE.txt
| file.

Jetty 6.1.26 includes the following in its NOTICE file:

=====
| Jetty Web Container
| Copyright 1995-2009 Mort Bay Consulting Pty Ltd

=====
|
| The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
| unless otherwise noted. It is licensed under the apache 2.0
| license.

|
| The javax.servlet package used by Jetty is copyright
| Sun Microsystems, Inc and Apache Software Foundation. It is
| distributed under the Common Development and Distribution License.
| You can obtain a copy of the license at
| <https://glassfish.dev.java.net/public/CDDLv1.0.html>.

|
| The UnixCrypt.java code ~Implements the one way cryptography used by
| Unix systems for simple password protection. Copyright 1996 Aki Yoshida,
| modified April 2001 by Iris Van den Broeke, Daniel Deville.
| Permission to use, copy, modify and distribute UnixCrypt
| for non-commercial or commercial purposes and without fee is
| granted provided that the copyright notice appears in all copies.
|
| The default JSP implementation is provided by the Glassfish JSP engine
| from project Glassfish <https://glassfish.dev.java.net>. Copyright 2005
| Sun Microsystems, Inc. and portions Copyright Apache Software Foundation.
|
| Some portions of the code are Copyright:
| 2006 Tim Venum
| 1999 Jason Gilbert.
|
| The jboss integration module contains some LGPL code.
| [JBoss INTEGRATION IS NOT INCLUDED IN AVRO TOOLS.]
|
| The win32 Java Service Wrapper (v3.2.3) is Copyright (c) 1999, 2006
| Tanuki Software, Inc. and 2001 Silver Egg Technology. It is
| covered by an open license which is viewable at
| <http://svn.codehaus.org/jetty/jetty/branches/jetty-6.1/extras/win32service/LICENSE.txt>
| [WIN32 WRAPPER IS NOT INCLUDED IN AVRO TOOLS.]

Netty 3.5.13.Final includes the following in its NOTICE file:

|
| The Netty Project
| =====

|
| Please visit the Netty web site for more information:
|
| * <https://netty.io/>
|
| Copyright 2011 The Netty Project
|
| The Netty Project licenses this file to you under the Apache License,
| version 2.0 (the "License"); you may not use this file except in compliance
| with the License. You may obtain a copy of the License at:
|
| <https://www.apache.org/licenses/LICENSE-2.0>
|
| Unless required by applicable law or agreed to in writing, software
| distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
| WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
| License for the specific language governing permissions and limitations
| under the License.

| Also, please refer to each LICENSE.<component>.txt file, which is located in
| the 'license' directory of the distribution file, for the license terms of the
| components that this product depends on.

| -----
| This product contains the extensions to Java Collections Framework which has
| been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

| * LICENSE:
| * license/LICENSE.jsr166y.txt (Public Domain)
| * HOMEPAGE:
| * <http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/>
| * <http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jboss/cache/experimental/jsr166/>

| This product contains a modified version of Robert Harder's Public Domain
| Base64 Encoder and Decoder, which can be obtained at:

| * LICENSE:
| * license/LICENSE.base64.txt (Public Domain)
| * HOMEPAGE:
| * <http://iharder.sourceforge.net/current/java/base64/>

| This product contains a modified version of 'JZlib', a re-implementation of
| zlib in pure Java, which can be obtained at:

| * LICENSE:
| * license/LICENSE.jzlib.txt (BSD Style License)
| * HOMEPAGE:
| * <http://www.jcraft.com/jzlib/>

| This product optionally depends on 'Protocol Buffers', Google's data
| interchange format, which can be obtained at:

| * LICENSE:
| * license/LICENSE.protobuf.txt (New BSD License)
| * HOMEPAGE:
| * <https://code.google.com/p/protobuf/>

| This product optionally depends on 'SLF4J', a simple logging facade for Java,
| which can be obtained at:

| * LICENSE:
| * license/LICENSE.slf4j.txt (MIT License)
| * HOMEPAGE:
| * <https://www.slf4j.org/>

| This product optionally depends on 'Apache Commons Logging', a logging

| framework, which can be obtained at:

|

| * LICENSE:

| * license/LICENSE.commons-logging.txt (Apache License 2.0)

| * HOMEPAGE:

| * <https://commons.apache.org/logging/>

|

| This product optionally depends on 'Apache Log4J', a logging framework,

| which can be obtained at:

|

| * LICENSE:

| * license/LICENSE.log4j.txt (Apache License 2.0)

| * HOMEPAGE:

| * <https://logging.apache.org/log4j/>

|

| This product optionally depends on 'JBoss Logging', a logging framework,

| which can be obtained at:

|

| * LICENSE:

| * license/LICENSE.jboss-logging.txt (GNU LGPL 2.1)

| * HOMEPAGE:

| * <https://anonsvn.jboss.org/repos/common/common-logging-spi/>

|

| [JBASS LOGGING IS NOT INCLUDED IN AVRO TOOLS.]

|

| This product optionally depends on 'Apache Felix', an open source OSGi

| framework implementation, which can be obtained at:

|

| * LICENSE:

| * license/LICENSE.felix.txt (Apache License 2.0)

| * HOMEPAGE:

| * <https://felix.apache.org/>

|

| [FELIX IS NOT INCLUDED IN AVRO TOOLS.]

|

| This product optionally depends on 'Webbit', a Java event based

| WebSocket and HTTP server:

|

| * LICENSE:

| * license/LICENSE.webbit.txt (BSD License)

| * HOMEPAGE:

| * <https://github.com/joewalnes/webbit>

|

| [WEBBIT IS NOT INCLUDED IN AVRO TOOLS.]

Apache Commons CLI includes the following in its NOTICE file:

| Apache Commons CLI

| Copyright 2001-2009 The Apache Software Foundation
|
| This product includes software developed by
| The Apache Software Foundation (<https://www.apache.org/>).

Apache Commons logging includes the following in its NOTICE file:

| Apache Commons Logging
| Copyright 2003-2007 The Apache Software Foundation
|
| This product includes software developed by
| The Apache Software Foundation (<https://www.apache.org/>).

Apache Commons HttpClient includes the following in its NOTICE file:

| Apache Jakarta HttpClient
| Copyright 1999-2007 The Apache Software Foundation
|
| This product includes software developed by
| The Apache Software Foundation (<https://www.apache.org/>).

Apache Hadoop includes the following in its NOTICE file:

| This product includes software developed by The Apache Software
| Foundation (<https://www.apache.org/>).

Apache Commons IO includes the following in its NOTICE file:

| Apache Commons IO
| Copyright 2002-2012 The Apache Software Foundation
|
| This product includes software developed by
| The Apache Software Foundation (<https://www.apache.org/>).

Apache Commons Net includes the following in its NOTICE file:

| Apache Commons Net
| Copyright 2001-2012 The Apache Software Foundation
|
| This product includes software developed by
| The Apache Software Foundation (<https://www.apache.org/>).

Apache Log4j includes the following in its NOTICE file:

| Apache log4j
| Copyright 2010 The Apache Software Foundation
|
| This product includes software developed at

| The Apache Software Foundation (<https://www.apache.org/>).

Apache Commons configuration includes the following in its NOTICE file:

| Apache Commons Configuration

| Copyright 2001-2008 The Apache Software Foundation

|

| This product includes software developed by

| The Apache Software Foundation (<https://www.apache.org/>).

Apache Commons digester includes the following in its NOTICE file:

| Apache Jakarta Commons Digester

| Copyright 2001-2006 The Apache Software Foundation

|

| This product includes software developed by

| The Apache Software Foundation (<https://www.apache.org/>).

Apache Commons beanutils includes the following in its NOTICE file:

| Apache Commons BeanUtils

| Copyright 2000-2008 The Apache Software Foundation

|

| This product includes software developed by

| The Apache Software Foundation (<https://www.apache.org/>).

Apache Directory includes the following in its NOTICE file:

| ApacheDS

| Copyright 2003-2015 The Apache Software Foundation

|

| This product includes software developed at

| The Apache Software Foundation (<https://www.apache.org/>).

Apache Zookeeper includes the following in its NOTICE file:

| Apache ZooKeeper

| Copyright 2009-2014 The Apache Software Foundation

|

| This product includes software developed at

| The Apache Software Foundation (<https://www.apache.org/>).

Apache Avro

Copyright 2010-2015 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (<https://www.apache.org/>).

This library was original developed by Yann Kerherve with the following

copyright notice:

| Copyright (C) 2010 Yann Kerherve. All rights reserved.
| Apache Avro
| Copyright 2010 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

Based upon the representations of upstream licensors, it is understood that portions of the mapreduce API included in the Java implementation are licensed from various contributors under one or more contributor license agreements to Odiago, Inc. and were then contributed by Odiago to Apache Avro, which has now made them available under the Apache 2.0 license. The original file header text is:

| Licensed to Odiago, Inc. under one or more contributor license
| agreements. See the NOTICE file distributed with this work for
| additional information regarding copyright ownership. Odiago, Inc.
| licenses this file to you under the Apache License, Version 2.0
| (the "License"); you may not use this file except in compliance
| with the License. You may obtain a copy of the License at
|
| <https://www.apache.org/licenses/LICENSE-2.0>
|
| Unless required by applicable law or agreed to in writing, software
| distributed under the License is distributed on an "AS IS" BASIS,
| WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
| implied. See the License for the specific language governing
| permissions and limitations under the License.

The Odiago NOTICE at the time of the contribution:

| This product includes software developed by Odiago, Inc.
| (<https://www.wibidata.com/>).
| Apache Avro
| Copyright 2010-2022 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

Based upon the representations of upstream licensors, it is understood that portions of the mapreduce API included in the Java implementation are licensed from various contributors under one or more contributor license agreements to Odiago, Inc. and were then contributed by Odiago to Apache Avro, which has now made them available under the Apache 2.0 license. The original file header text is:

| Licensed to Odiago, Inc. under one or more contributor license
| agreements. See the NOTICE file distributed with this work for
| additional information regarding copyright ownership. Odiago, Inc.
| licenses this file to you under the Apache License, Version 2.0
| (the "License"); you may not use this file except in compliance
| with the License. You may obtain a copy of the License at
|
| <https://www.apache.org/licenses/LICENSE-2.0>
|
| Unless required by applicable law or agreed to in writing, software
| distributed under the License is distributed on an "AS IS" BASIS,
| WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
| implied. See the License for the specific language governing
| permissions and limitations under the License.

|-----
| This product includes software developed by The Docsy Authors.
| (<https://www.docsy.dev/>).
|

| This product includes software developed at
| The Apache Software Foundation (<https://www.apache.org/>).
|

| See also the file LICENSE.txt
|

|-----
| The purpose of this NOTICE.txt file is to contain notices that are
| required by the copyright owner and their license. Some of the
| accompanying products have an attribution requirement, so see below.
| Other accompanying products do not require attribution, so are not listed.
|

|-----

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

License for msinttypes.h and msstdint.h used in the C implementation:

Source from:

<https://code.google.com/p/msinttypes/downloads/detail?name=msinttypes-r26.zip>

Copyright (c) 2006-2008 Alexander Chemeris

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are met:

|

| 1. Redistributions of source code must retain the above copyright notice,
| this list of conditions and the following disclaimer.

|

| 2. Redistributions in binary form must reproduce the above copyright
| notice, this list of conditions and the following disclaimer in the
| documentation and/or other materials provided with the distribution.

|

| 3. The name of the author may be used to endorse or promote products
| derived from this software without specific prior written permission.

|

| THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED
| WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
| MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
| EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
| SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
| PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;
| OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
| WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
| OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
| ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License for st.c and st.h used in the C implementation:

| This is a public domain general purpose hash table package written by
| Peter Moore @ UCB.

License for Diredt API for Microsoft Visual Studio used in the C implementation:

Source from:

<http://www.softgalleria.net/download/diredt/diredt-1.11.zip>

Copyright (C) 2006 Toni Ronkko

| Permission is hereby granted, free of charge, to any person obtaining
| a copy of this software and associated documentation files (the
| ``Software"), to deal in the Software without restriction, including
| without limitation the rights to use, copy, modify, merge, publish,
| distribute, sublicense, and/or sell copies of the Software, and to
| permit persons to whom the Software is furnished to do so, subject to
| the following conditions:

|

| The above copyright notice and this permission notice shall be included
| in all copies or substantial portions of the Software.

|
| THE SOFTWARE IS PROVIDED ``AS IS'', WITHOUT WARRANTY OF ANY KIND, EXPRESS
| OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
| MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
| IN NO EVENT SHALL TONI RONKKO BE LIABLE FOR ANY CLAIM, DAMAGES OR
| OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
| ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
| OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed

with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

License for portions of idl.jj in the Java compiler implementation:

Portions of idl.jj were modeled after the example Java 1.5 parser included with JavaCC. For those portions:

Copyright (c) 2006, Sun Microsystems, Inc.
All rights reserved.

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are met:

- | * Redistributions of source code must retain the above copyright notice,
| this list of conditions and the following disclaimer.
- | * Redistributions in binary form must reproduce the above copyright
| notice, this list of conditions and the following disclaimer in the
| documentation and/or other materials provided with the distribution.
- | * Neither the name of the Sun Microsystems, Inc. nor the names of its
| contributors may be used to endorse or promote products derived from
| this software without specific prior written permission.

| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
| AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
| IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
| ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
| LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
| CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
| SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
| INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN

| CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
| ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
| THE POSSIBILITY OF SUCH DAMAGE.

1.10 ginkgo 1.16.4

1.10.1 Available under license :

The MIT License (MIT)

Copyright (c) 2016 Yasuhiro Matsumoto

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) Yasuhiro MATSUMOTO <mattn.jp@gmail.com>

MIT License (Expat)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2013-2014 Onsi Fakhouri

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.11 docker-java-transport 3.2.13

1.11.1 Available under license :

Apache-2.0

1.12 jetbrains-annotations 13.0

1.12.1 Available under license :

Copyright 2000-2012 JetBrains s.r.o.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.13 golang-appengine v1.6.7

1.13.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.14 joda-time v2.10.2

1.14.1 Available under license :

=====

= NOTICE file corresponding to section 4d of the Apache License Version 2.0 =

=====

This product includes software developed by
Joda.org (<https://www.joda.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.15 Iz4-and-xxhash 1.7.1

1.15.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and

limitations under the License.

1.16 github.com/xiaomi/pegasus-go-client 20210427-snapshot-f3b6b08b

1.16.1 Available under license :

Apache Pegasus

Copyright 2022 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

src/rdsn/** - MIT License

The MIT License (MIT)

Copyright (c) 2015 Microsoft Corporation

-- Robust Distributed System Nucleus (rDSN) --

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

src/shell/linenoise/* - BSD-2-Clause License

Copyright (c) 2010-2014, Salvatore Sanfilippo <antirez at gmail dot com>
Copyright (c) 2010-2013, Pieter Noordhuis <pcnoordhuis at gmail dot com>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

src/shell/sds/* - BSD-2-Clause License

Copyright (c) 2006-2015, Salvatore Sanfilippo <antirez at gmail dot com>
Copyright (c) 2015, Oran Agra
Copyright (c) 2015, Redis Labs, Inc
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Redis nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

src/shell/argh.h - BSD-3-Clause License

Copyright (c) 2016, Adi Shavit
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN

CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

src/rdsn/include/dsn/utility/smart_pointers.h - Apache 2.0 License
src/rdsn/include/dsn/utility/string_view.h
src/rdsn/include/dsn/utility/absl/base/internal/invoke.h
src/rdsn/include/dsn/utility/absl/utility/utility.h
src/rdsn/src/utls/memutil.h
src/rdsn/src/utls/string_view.cpp
src/rdsn/src/utls/test/memutil_test.cpp
src/rdsn/src/utls/test/smart_pointers_test.cpp
src/rdsn/src/utls/test/string_view_test.cpp

Copyright 2017 The Abseil Authors.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

src/rdsn/src/utls/safe_strerror_posix.cpp - BSD-3-Clause License

Copyright (c) 2006-2009 The Chromium Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from

this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

src/rdsn/include/dsn/utility/TokenBucket.h - Apache License, Version 2.0
src/rdsn/src/utills/test/TokenBucketTest.cpp
src/rdsn/src/utills/test/TokenBucketTest.h

Copyright (c) Facebook, Inc. and its affiliates.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

src/rdsn/include/dsn/utility/hpc_locks/autoreseteventcondvar.h - zlib License
src/rdsn/include/dsn/utility/hpc_locks/rwlock.h
src/rdsn/include/dsn/utility/hpc_locks/autoresetevent.h
src/rdsn/include/dsn/utility/hpc_locks/sema.h
src/rdsn/include/dsn/utility/hpc_locks/bitfield.h
src/rdsn/include/dsn/utility/hpc_locks/benaphore.h

Copyright (c) 2015 Jeff Preshing

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgement in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

src/rdsn/src/runtime/build_config.h - BSD-3-Clause License
src/rdsn/src/utills/test/autoref_ptr_test.cpp

Copyright (c) 2012 The Chromium Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

cmake_modules/FindRT.cmake - BSD-3-Clause license

cmake_modules/FindDL.cmake

Copyright (c) 2010-2011, Rob Jansen

To the extent that a federal employee is an author of a portion of this software or a derivative work thereof, no copyright is claimed by the United States Government, as represented by the Secretary of the Navy ("GOVERNMENT") under Title 17, U.S. Code. All Other Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the names of the copyright owners nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. GOVERNMENT ALLOWS FREE USE OF THIS SOFTWARE IN ITS "AS IS" CONDITION AND DISCLAIMS ANY LIABILITY OF ANY KIND FOR ANY DAMAGES WHATSOEVER RESULTING FROM THE USE OF THIS SOFTWARE.

scripts/run-clang-format.py - MIT License

MIT License

Copyright (c) 2017 Guillaume Papin

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

src/rdsn/include/dsn/utility/safe_strerror_posix.h - BSD-3-Clause License

Copyright (c) 2011 The Chromium Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

/*

* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.

*/

Copyright (c) 2010-2014, Salvatore Sanfilippo <antirez at gmail dot com>

Copyright (c) 2010-2013, Pieter Noordhuis <pnoordhuis at gmail dot com>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache Pegasus is an effort undergoing incubation at The Apache Software Foundation (ASF), sponsored by the Apache Incubator. Incubation is required of all newly accepted projects until a further review indicates that the infrastructure, communications, and decision making process have stabilized in a manner consistent with other successful ASF projects. While incubation status is not necessarily a reflection of the completeness or stability of the code, it does indicate that the project has yet to be fully endorsed by the ASF.

Some of the incubating project's releases may not be fully compliant with ASF policy. For example, releases may have incomplete or un-reviewed licensing conditions. What follows is a list of known issues the project is currently aware of (note that this list, by definition, is likely to be incomplete):

- * Releases may have incomplete licensing conditions.
- * Most of the top contributors have signed an ICLA and we are working on updating the headers.

If you are planning to incorporate this work into your product/project, please be aware that you will need to conduct a thorough licensing review to determine the overall implications of including this work. For the current status of this project through the Apache Incubator visit: <https://incubator.apache.org/projects/pegasus.html>

Copyright (c) 2006-2014, Salvatore Sanfilippo <antirez at gmail dot com>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR

ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.17 text v0.4.0

1.17.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they

know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an

exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological

measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no

further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install

modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in

reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license

you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of

the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
{one line to give the program's name and a brief idea of what it does.}
Copyright (C) {year} {name of author}
```

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
{project} Copyright (C) {year} {fullname}
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands

might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

1.18 kr-pretty 0.1.0

1.18.1 Available under license :

The MIT License (MIT)

Copyright 2012 Keith Rarick

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.19 aws-java-sdk-::-core-::-protocols-::-json- utils 2.17.122

1.19.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed
with the accompanying software if such terms are included in the directory containing the accompanying software.
Such other license terms will then apply in lieu of the terms of the software license above.

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by
Amazon Technologies, Inc (<http://www.amazon.com/>).

THIRD PARTY COMPONENTS

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:
Apache Commons Lang
Copyright 2001-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

1.20 junit-platform-junit-platform-engine 1.8.2

1.20.1 Available under license :

Eclipse Public License - v 2.0

=====

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. Definitions

Contribution means:

* **a)** in the case of the initial Contributor, the initial content Distributed under this Agreement, and

* **b)** in the case of each subsequent Contributor:

* **i)** changes to the Program, and

* **ii)** additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of **a)** distributing or **b)** making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. Grant of Rights

****a)**** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

****b)**** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

****c)**** Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

****d)**** Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

****e)**** Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. Requirements

****3.1)**** If a Contributor Distributes the Program in any form, then:

****a)**** the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

****b)**** the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

****i)**** effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

****ii)**** effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

****iii)**** does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

****iv)**** requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

****3.2**** When the Program is Distributed as Source Code:

* ****a)**** it must be made available under this Agreement, or if the Program ****i)**** is combined with other material in a separate file or files made available under a Secondary License, and ****ii)**** the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

* ****b)**** a copy of this Agreement must be included with each copy of the Program.

****3.3**** Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: ****a)**** promptly notify the Commercial Contributor in writing of such claim, and ****b)**** allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

> This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Open Source Licenses

=====

This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the subcomponent's license, as noted in the LICENSE-<subcomponent>.md files.

1.21 testcontainers-core 1.17.3

1.21.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.22 apache-http-client 4.5.13

1.22.1 Available under license :

Apache HttpComponents Client
Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner

or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

=====

This project includes Public Suffix List copied from
<https://publicsuffix.org/list/effective_tld_names.dat>
licensed under the terms of the Mozilla Public License, v. 2.0

Full license text: <<http://mozilla.org/MPL/2.0/>>

Mozilla Public License Version 2.0

=====

1. Definitions

1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"

means Covered Software of a particular Contributor.

1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"

means

(a) that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form"

means any form of the work other than Source Code Form.

1.7. "Larger Work"

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. "License"

means this document.

1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. "Modifications"

means any of the following:

(a) any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

(b) any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"

means the form of the work preferred for making modifications.

1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- (a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- (b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution

become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License.

Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- (a) for any code that a Contributor has removed from Covered Software;
or
- (b) for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- (c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted

in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- (a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- (b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of

the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions,

counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

```
*
*
* 6. Disclaimer of Warranty
* -----
*
* Covered Software is provided under this License on an "as is"
* basis, without warranty of any kind, either expressed, implied, or
* statutory, including, without limitation, warranties that the
* Covered Software is free of defects, merchantable, fit for a
* particular purpose or non-infringing. The entire risk as to the
* quality and performance of the Covered Software is with You.
* Should any Covered Software prove defective in any respect, You
* (not any Contributor) assume the cost of any necessary servicing,
* repair, or correction. This disclaimer of warranty constitutes an
* essential part of this License. No use of any Covered Software is
* authorized under this License except under this disclaimer.
*
*
```

```
*
*
* 7. Limitation of Liability
* -----
*
* Under no circumstances and under no legal theory, whether tort
* (including negligence), contract, or otherwise, shall any
* Contributor, or anyone who distributes Covered Software as
* permitted above, be liable to You for any direct, indirect,
* special, incidental, or consequential damages of any character
* including, without limitation, damages for lost profits, loss of
* goodwill, work stoppage, computer failure or malfunction, or any
* and all other commercial damages or losses, even if such party
* shall have been informed of the possibility of such damages. This
* limitation of liability shall not apply to liability for death or
* personal injury resulting from such party's negligence to the
* extent applicable law prohibits such limitation. Some
* jurisdictions do not allow the exclusion or limitation of
* incidental or consequential damages, so this exclusion and
```

* limitation may not apply to You. *
* *

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

1.23 zstd-jni 1.4.9-1

1.23.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Automatic-Module-Name: com.github.luben.zstd_jni

Bnd-LastModified: 1615191527618

Bundle-Description: JNI bindings for Zstd native library that provides fast and high compression lossless algorithm for Java and all JVM languages.

Bundle-License: <https://opensource.org/licenses/BSD-2-Clause>;description=BSD 2-Clause License

Bundle-ManifestVersion: 2

Bundle-Name: zstd-jni

Bundle-NativeCode: aix/ppc64/libzstd-jni.so;osname=AIX;processor=ppc64, darwin/x86_64/libzstd-jni.dylib;osname=MacOS;osname=MacOSX;processor=x86_64, darwin/aarch64/libzstd-jni.dylib;osname=MacOS;osname=MacOSX;processor=aarch64, freebsd/amd64/libzstd-jni.so;osname=FreeBSD;proce

ssor=amd64, freebsd/i386/libzstd-jni.so;osname=FreeBSD;processor=i386
, linux/aarch64/libzstd-jni.so;osname=Linux;processor=aarch64, linux/
amd64/libzstd-jni.so;osname=Linux;processor=amd64, linux/arm/libzstd-
jni.so;osname=Linux;processor=arm, linux/i386/libzstd-jni.so;osname=L
inux;processor=i386, linux/mips64/libzstd-jni.so;osname=Linux;process
or=mips64, linux/ppc64/libzstd-jni.so;osname=Linux;processor=ppc64, l
inux/ppc64le/libzstd-jni.so;osname=Linux;processor=ppc64le, linux/s39
0x/libzstd-jni.so;osname=Linux;processor=s390x, win/amd64/libzstd-jni
.dll;osname=Win32;processor=amd64, win/x86/libzstd-jni.dll;osname=Win
32;processor=x86
Bundle-SymbolicName: com.github.luben.zstd-jni
Bundle-Vendor: com.github.luben
Bundle-Version: 1.4.9.1
Created-By: 1.8.0_275 (Debian)
Export-Package: com.github.luben.zstd;version="1.4.9.1",com.github.lub
en.zstd.util;version="1.4.9.1"
Implementation-Title: zstd-jni
Implementation-Vendor: com.github.luben
Implementation-Vendor-Id: com.github.luben
Implementation-Version: 1.4.9-1
Import-Package: org.osgi.framework;resolution:=optional
Private-Package: linux.amd64,linux.i386,linux.aarch64,linux.arm,linux.
ppc64,linux.ppc64le,linux.mips64,linux.s390x,aix.ppc64,darwin.x86_64,
darwin.aarch64,win.amd64,win.x86,freebsd.amd64,freebsd.i386
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"
Specification-Title: zstd-jni
Specification-Vendor: com.github.luben
Specification-Version: 1.4.9-1
Tool: Bnd-4.0.0.201805111645

Found in path(s):

* /opt/cola/permits/1183892379_1627494642.79/0/zstd-jni-1-4-9-1-1-jar/META-INF/MANIFEST.MF

1.24 apache-log4j 2.17.1

1.24.1 Available under license :

Apache Log4j Core

Copyright 1999-2012 Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (<http://www.apache.org/>).

ResolverUtil.java

Copyright 2005-2006 Tim Fennell

Apache License

Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

*

* Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache license, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the license for the specific language governing permissions and limitations under the license.

*/

1.25 netty-reactive-streams-implementation

2.0.5

1.25.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bundle-Description: Reactive streams implementation for Netty.

Automatic-Module-Name: com.typesafe.netty.core

Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>

Bundle-SymbolicName: com.typesafe.netty.reactive-streams

Built-By: marcospereira

Bnd-LastModified: 1602622953847

Bundle-ManifestVersion: 2

Bundle-DocURL: <http://typesafe.com/>

Bundle-Vendor: Typesafe

Import-Package: io.netty.channel;version="[4.1,5)",io.netty.util;version="[4.1,5)",io.netty.util.concurrent;version="[4.1,5)",io.netty.util.internal;version="[4.1,5)",org.reactivestreams;version="[1.0,2)"

Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.7))"

Tool: Bnd-3.5.0.201709291849

Export-Package: com.typesafe.netty;uses:="io.netty.channel,io.netty.ut

il.concurrent.org.reactivestreams";version="2.0.5"
Bundle-Name: Netty Reactive Streams Implementation
Bundle-Version: 2.0.5
Created-By: Apache Maven Bundle Plugin
Build-Jdk: 1.8.0_181

Found in path(s):

* /opt/cola/permits/1128619582_1649176967.68/0/netty-reactive-streams-2-0-5-jar/META-INF/MANIFEST.MF

1.26 wire-schema 3.7.1

1.26.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2019 Square, Inc.  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

Found in path(s):

```
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-  
jar/jvmMain/com/squareup/wire/schema/CoreLoader.kt  
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-  
jar/commonMain/com/squareup/wire/schema/Loader.kt  
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-  
jar/commonMain/com/squareup/wire/schema/SemVer.kt  
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-  
jar/jvmMain/com/squareup/wire/schema/internal/UtilJVM.kt  
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-  
jar/commonMain/com/squareup/wire/schema/FileLinker.kt
```

No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2017 Square, Inc.  
*  
* Licensed under the Apache License, Version 2.0 (the "License");
```

* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/internal/parser/OptionReader.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2014 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/internal/parser/FieldElement.kt

* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/internal/parser/EnumConstantElement.kt

* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/internal/parser/RpcElement.kt

* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/internal/parser/OneOfElement.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2021 Square Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/internal/SchemaEncoder.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2016 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/EnclosingType.kt

* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/internal/parser/SyntaxReader.kt

* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/internal/parser/ReservedElement.kt

* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/internal/parser/GroupElement.kt

* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/Reserved.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2020 Square, Inc.

*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/Multimap.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/internal/DagChecker.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/jvmMain/com/squareup/wire/schema/internal/JvmLanguages.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/internal/TypeMover.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/jvmMain/com/squareup/wire/schema/Multimap.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/jvmMain/com/squareup/wire/schema/EmittingRules.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/jvmMain/com/squareup/wire/schema/ClaimedDefinitions.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/CoreLoader.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/SyntaxRules.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/CycleChecker.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/ErrorCollector.kt

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
*/

- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/internal/parser/TypeElement.kt
- * /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/internal/Util.kt
- * /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/internal/parser/EnumElement.kt
- * /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/internal/parser/ExtensionsElement.kt
- * /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/internal/parser/OptionElement.kt
- * /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/internal/parser/MessageElement.kt
- * /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/internal/parser/ServiceElement.kt
- * /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/internal/parser/ProtoFileElement.kt
- * /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/internal/parser/ProtoParser.kt
- * /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-jar/commonMain/com/squareup/wire/schema/internal/parser/ExtendElement.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2015 Square, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

- * /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-

jar/commonMain/com/squareup/wire/schema/Extensions.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-
jar/commonMain/com/squareup/wire/schema/Location.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-
jar/commonMain/com/squareup/wire/schema/OneOf.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-
jar/commonMain/com/squareup/wire/schema/MessageType.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-
jar/commonMain/com/squareup/wire/schema/Rpc.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-
jar/commonMain/com/squareup/wire/schema/SchemaProtoAdapterFactory.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-
jar/commonMain/com/squareup/wire/schema/Type.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-
jar/commonMain/com/squareup/wire/schema/Extend.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-
jar/commonMain/com/squareup/wire/schema/Linker.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-
jar/commonMain/com/squareup/wire/schema/Field.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-
jar/commonMain/com/squareup/wire/schema/Service.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-
jar/commonMain/com/squareup/wire/schema/Schema.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-
jar/commonMain/com/squareup/wire/schema/SchemaException.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-
jar/commonMain/com/squareup/wire/schema/ProtoType.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-
jar/commonMain/com/squareup/wire/schema/PruningRules.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-
jar/commonMain/com/squareup/wire/schema/MarkSet.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-
jar/commonMain/com/squareup/wire/schema/Options.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-
jar/commonMain/com/squareup/wire/schema/ProtoFile.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-
jar/commonMain/com/squareup/wire/schema/Pruner.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-
jar/commonMain/com/squareup/wire/schema/EnumConstant.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-
jar/commonMain/com/squareup/wire/schema/ProtoMember.kt
* /opt/cola/permits/1526005825_1673041453.79524/0/wire-schema-3-7-1-sources-
jar/commonMain/com/squareup/wire/schema/EnumType.kt

1.27 aws-sdk-for-go v1.44.140

1.27.1 Available under license :

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.28 wire-protocol-buffer-java-generator 3.7.1

1.28.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2015 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1526005763_1673463505.600457/0/wire-java-generator-3-7-1-sources-
jar/com/squareup/wire/java/JavaGenerator.java
```

1.29 go-tomb-tomb 20150422-snapshot-dd632973

1.29.1 Available under license :

tomb - support for clean goroutine termination in Go.

Copyright (c) 2010-2011 - Gustavo Niemeyer <gustavo@niemeyer.net>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.30 spf13-cast v1.4.1

1.30.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Steve Francia

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.31 aws-sdk-for-java 1.12.349

1.31.1 Available under license :

Apache License

Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section)

patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

AWS SDK for Java

Copyright 2010-2014 Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by
Amazon Technologies, Inc (<http://www.amazon.com/>).

THIRD PARTY COMPONENTS

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.

The licenses for these third party components are included in LICENSE.txt

1.32 opencensus-api 0.28.0

1.32.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2018, OpenCensus Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

```
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
* API for resource information population.
*
* <p>The resource library primarily defines a type "Resource" that captures information about the
* entity for which stats or traces are recorded. For example, metrics exposed by a Kubernetes
* container can be linked to a resource that specifies the cluster, namespace, pod, and container
* name.
*
* <p>Two environment variables are used to populate resource information:
*
* <ul>
* <li>OC_RESOURCE_TYPE: A string that describes the type of the resource prefixed by a domain
* namespace. Leading and trailing whitespaces are trimmed. e.g. "kubernetes.io/container".
* <li>OC_RESOURCE_LABELS: A comma-separated list of labels describing the source in more detail,
* e.g. "key1=val1,key2=val2". The allowed character set is appropriately constrained.
* </ul>
*
* <p>Type, label keys, and label values MUST contain only printable ASCII (codes between 32 and
* 126, inclusive) and less than 256 characters. Type and label keys MUST have a length greater than
* zero. They SHOULD start with a domain name and separate hierarchies with / characters, e.g.
* k8s.io/namespace/name.
*
* <p>WARNING: Currently all the public classes under this package are marked as {@link
* io.opencensus.common.ExperimentalApi}. DO NOT USE except for experimental purposes.
*
* <p>Please see
* https://github.com/census-instrumentation/opencensus-specs/blob/master/resource/Resource.md for
* more details.
*/
```

Found in path(s):

```
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/resource/package-info.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2018, OpenCensus Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
```

- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/metrics/DerivedLongGauge.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/BigendianEncoding.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/metrics/LabelKey.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/internal/Utils.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/common/ServerStatsDeserializationException.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/metrics/export/MetricProducer.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/common/package-info.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/metrics/export/Value.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/Tracestate.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/metrics/export/MetricProducerManager.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/metrics/export/Point.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/common/ServerStats.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/common/ToLongFunction.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/metrics/DoubleGauge.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/metrics/MetricsComponent.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/package-info.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/metrics/LongGauge.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-

jar/io/opencensus/metrics/DerivedDoubleGauge.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/metrics/export/Distribution.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/internal/DefaultVisibilityForTesting.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/metrics/Metrics.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/metrics/export/TimeSeries.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/metrics/export/ExportComponent.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/common/ToDoubleFunction.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/stats/package-info.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/resource/Resource.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/trace/MessageEvent.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/metrics/export/Summary.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/trace/BaseMessageEvent.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/metrics/LabelValue.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/metrics/export/Metric.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/common/ServerStatsEncoding.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/trace/internal/BaseMessageEventUtils.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/common/ServerStatsFieldEnums.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/metrics/export/MetricDescriptor.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/metrics/MetricRegistry.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2019, OpenCensus Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/tags/propagation/TagContextTextFormat.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/metrics/DerivedDoubleCumulative.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/metrics/DoubleCumulative.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/metrics/LongCumulative.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/metrics/data/AttachmentValue.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/metrics/DerivedLongCumulative.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/metrics/data/Exemplar.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/tags/TagMetadata.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/metrics/MetricOptions.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/metrics/data/package-info.java

No license file was found, but licenses were detected in source scan.

/*

- * Copyright 2016-17, OpenCensus Authors
- *
- * Licensed under the Apache License, Version 2.0 (the "License");
- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-

jar/io/opencensus/trace/Tracing.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/stats/ViewData.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/internal/StringUtils.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/stats/View.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/trace/AttributeValue.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/stats/Measurement.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/trace/Tracer.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/trace/NetworkEvent.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/stats/ViewManager.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/trace/Span.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/stats/Stats.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/trace/Status.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/common/Timestamp.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/common/OpenCensusLibraryInformation.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/trace/TraceId.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/internal/Provider.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/common/Function.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/trace/BlankSpan.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/trace/SpanContext.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/trace/TraceComponent.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/common/Duration.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright 2017, OpenCensus Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

- * You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/SpanBuilder.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/tags/propagation/TagContextDeserializationException.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/unsafe/ContextUtils.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/tags/TagsComponent.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/tags/TagValue.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/config/TraceConfig.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/propagation/SpanContextParseException.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/Sampler.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/common/ExperimentalApi.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/TraceOptions.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/export/SpanData.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/samplers/Samplers.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/common/Functions.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/stats/Measure.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/tags/Tags.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/tags/TagKey.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/internal/ZeroTimeClock.java
- * /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/stats/BucketBoundaries.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/EndSpanOptions.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/export/ExportComponent.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/stats/AggregationData.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/tags/Tag.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/export/SampledSpanStore.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/common/Internal.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/internal/NoopScope.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/tags/InternalUtils.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/export/SpanExporter.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/propagation/TextFormat.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/stats/StatsCollectionState.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/samplers/NeverSampleSampler.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/tags/propagation/TagPropagationComponent.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/stats/StatsRecorder.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/tags/unsafe/ContextUtils.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/tags/NoopTags.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/stats/StatsComponent.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/tags/TagContext.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/stats/MeasureMap.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/tags/Tagger.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/stats/NoopStats.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/export/RunningSpanStore.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/SpanId.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/propagation/BinaryFormat.java

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/stats/Aggregation.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/tags/package-info.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/tags/propagation/TagContextSerializationException.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/Annotation.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/tags/TaggingState.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/Link.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/CurrentSpanUtils.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/propagation/PropagationComponent.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/common/Clock.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/samplers/AlwaysSampleSampler.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/config/TraceParams.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/tags/TagContextBuilder.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/internal/package-info.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/tags/propagation/TagContextBinarySerializer.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/common/NonThrowingCloseable.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/common/Scope.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/trace/samplers/ProbabilitySampler.java
* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-jar/io/opencensus/common/TimeUtils.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2018, OpenCensus Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

```

* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
* This package describes the Metrics data model. Metrics are a data model for what stats exporters
* take as input. This data model may eventually become the wire format for metrics.
*
* <p>WARNING: Currently all the public classes under this package are marked as { @link
* io.opencensus.common.ExperimentalApi}. The classes and APIs under { @link io.opencensus.metrics}
* are likely to get backwards-incompatible updates in the future. DO NOT USE except for
* experimental purposes.
*
* <p>Please see
* https://github.com/census-instrumentation/opencensus-specs/blob/master/stats/Metrics.md and
* https://github.com/census-instrumentation/opencensus-proto/blob/master/opencensus/proto/stats/metrics/metrics.proto
* for more details.
*/

```

Found in path(s):

```

* /opt/cola/permits/1411867099_1662683857.9450252/0/opencensus-api-0-28-0-sources-1-
jar/io/opencensus/metrics/package-info.java

```

1.33 asm-analysis 9.1

1.33.1 Available under license :

No license file was found, but licenses were detected in source scan.

2011 INRIA, France Telecom

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. Neither the name of the copyright holders nor the names of its

* contributors may be used to endorse or promote products derived from

* this software without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):

* /opt/cola/permits/1183891316_1627493686.82/0/asm-analysis-9-1-sources-
jar/org/objectweb/asm/tree/analysis/package.html

No license file was found, but licenses were detected in source scan.

// All rights reserved.

// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:

// 1. Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.

// 2. Redistributions in binary form must reproduce the above copyright
// notice, this list of conditions and the following disclaimer in the
// documentation and/or other materials provided with the distribution.

// 3. Neither the name of the copyright holders nor the names of its
// this software without specific prior written permission.

Found in path(s):

* /opt/cola/permits/1183891316_1627493686.82/0/asm-analysis-9-1-sources-
jar/org/objectweb/asm/tree/analysis/BasicVerifier.java

* /opt/cola/permits/1183891316_1627493686.82/0/asm-analysis-9-1-sources-
jar/org/objectweb/asm/tree/analysis/Analyzer.java

* /opt/cola/permits/1183891316_1627493686.82/0/asm-analysis-9-1-sources-
jar/org/objectweb/asm/tree/analysis/SourceValue.java

* /opt/cola/permits/1183891316_1627493686.82/0/asm-analysis-9-1-sources-
jar/org/objectweb/asm/tree/analysis/SimpleVerifier.java

* /opt/cola/permits/1183891316_1627493686.82/0/asm-analysis-9-1-sources-
jar/org/objectweb/asm/tree/analysis/Interpreter.java

* /opt/cola/permits/1183891316_1627493686.82/0/asm-analysis-9-1-sources-
jar/org/objectweb/asm/tree/analysis/Value.java

* /opt/cola/permits/1183891316_1627493686.82/0/asm-analysis-9-1-sources-
jar/org/objectweb/asm/tree/analysis/AnalyzerException.java

* /opt/cola/permits/1183891316_1627493686.82/0/asm-analysis-9-1-sources-
jar/org/objectweb/asm/tree/analysis/Frame.java

* /opt/cola/permits/1183891316_1627493686.82/0/asm-analysis-9-1-sources-
jar/org/objectweb/asm/tree/analysis/BasicValue.java

* /opt/cola/permits/1183891316_1627493686.82/0/asm-analysis-9-1-sources-
jar/org/objectweb/asm/tree/analysis/Subroutine.java

* /opt/cola/permits/1183891316_1627493686.82/0/asm-analysis-9-1-sources-jar/org/objectweb/asm/tree/analysis/SmallSet.java

No license file was found, but licenses were detected in source scan.

// All rights reserved.

// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:

// 1. Redistributions of source code must retain the above copyright

// notice, this list of conditions and the following disclaimer.

// 2. Redistributions in binary form must reproduce the above copyright

// notice, this list of conditions and the following disclaimer in the

// documentation and/or other materials provided with the distribution.

// 3. Neither the name of the copyright holders nor the names of its

// this software without specific prior written permission.

/**

* Constructs a new { @link BasicInterpreter} for the latest ASM API version. <i>Subclasses must

* not use this constructor</i>. Instead, they must use the { @link #BasicInterpreter(int)}

* version.

*/

Found in path(s):

* /opt/cola/permits/1183891316_1627493686.82/0/asm-analysis-9-1-sources-jar/org/objectweb/asm/tree/analysis/BasicInterpreter.java

No license file was found, but licenses were detected in source scan.

// All rights reserved.

// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:

// 1. Redistributions of source code must retain the above copyright

// notice, this list of conditions and the following disclaimer.

// 2. Redistributions in binary form must reproduce the above copyright

// notice, this list of conditions and the following disclaimer in the

// documentation and/or other materials provided with the distribution.

// 3. Neither the name of the copyright holders nor the names of its

// this software without specific prior written permission.

/**

* Constructs a new { @link SourceInterpreter} for the latest ASM API version. <i>Subclasses must

* not use this constructor</i>. Instead, they must use the { @link #SourceInterpreter(int)}

* version.

*/

Found in path(s):

* /opt/cola/permits/1183891316_1627493686.82/0/asm-analysis-9-1-sources-jar/org/objectweb/asm/tree/analysis/SourceInterpreter.java

1.34 slf4j-api-module 1.7.32

1.34.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/**
 * Copyright (c) 2004-2011 QOS.ch
 * All rights reserved.
 *
 * Permission is hereby granted, free of charge, to any person obtaining
 * a copy of this software and associated documentation files (the
 * "Software"), to deal in the Software without restriction, including
 * without limitation the rights to use, copy, modify, merge, publish,
 * distribute, sublicense, and/or sell copies of the Software, and to
 * permit persons to whom the Software is furnished to do so, subject to
 * the following conditions:
 *
 * The above copyright notice and this permission notice shall be
 * included in all copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
 * EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
 * MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
 * NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
 * LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
 * OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
 * WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
 */
```

Found in path(s):

```
* /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-jar/org/slf4j/MDC.java
* /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-jar/org/slf4j/Logger.java
* /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-
jar/org/slf4j/impl/StaticMDCBinder.java
* /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-
jar/org/slf4j/spi/MarkerFactoryBinder.java
* /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-
jar/org/slf4j/helpers/BasicMarkerFactory.java
* /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-
jar/org/slf4j/helpers/SubstituteLogger.java
* /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-jar/org/slf4j/LoggerFactory.java
* /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-
jar/org/slf4j/helpers/BasicMDCAdapter.java
* /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-jar/org/slf4j/MarkerFactory.java
* /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-
jar/org/slf4j/helpers/SubstituteLoggerFactory.java
```

- * /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-jar/org/slf4j/Marker.java
- * /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-jar/org/slf4j/helpers/BasicMarker.java
- * /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-jar/org/slf4j/helpers/MarkerIgnoringBase.java
- * /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-jar/org/slf4j/helpers/NOPLogger.java
- * /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-jar/org/slf4j/spi/LocationAwareLogger.java
- * /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-jar/org/slf4j/helpers/NOPMDCAadapter.java
- * /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-jar/org/slf4j/helpers/NOPLoggerFactory.java
- * /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-jar/org/slf4j/impl/StaticMarkerBinder.java
- * /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-jar/org/slf4j/impl/StaticLoggerBinder.java
- * /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-jar/org/slf4j/spi/LoggerFactoryBinder.java
- * /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-jar/org/slf4j/IMarkerFactory.java
- * /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-jar/org/slf4j/ILoggerFactory.java
- * /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-jar/org/slf4j/helpers/Util.java
- * /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-jar/org/slf4j/helpers/MessageFormatter.java
- * /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-jar/org/slf4j/spi/MDCAdapter.java
- * /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-jar/org/slf4j/helpers/NamedLoggerBase.java
- * /opt/cola/permits/1267277703_1648511906.32/0/slf4j-api-1-7-32-sources-jar/org/slf4j/helpers/FormattingTuple.java

1.35 rxjava 3.1.3

1.35.1 Available under license :

Copyright (c) 2016-present, RxJava Contributors.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See

the License for the specific language governing permissions and limitations under the License.

/*

* Copyright (c) 2016-present, RxJava Contributors.

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in

* compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License is

* distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See

* the License for the specific language governing permissions and limitations under the License.

*/

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed

with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.36 kotlin 1.7.20

1.36.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
// Licensed under the Apache License, Version 2.0 (the "License");  
// you may not use this file except in compliance with the License.  
// You may obtain a copy of the License at  
//   http://www.apache.org/licenses/LICENSE-2.0  
// distributed under the License is distributed on an "AS-IS" BASIS,
```

Found in path(s):

```
* /opt/cola/permits/1526005960_1673041501.3165746/0/kotlin-1-7-20-1-tgz/package/kotlin.js
```

No license file was found, but licenses were detected in source scan.

```
{"version":3,"file":"kotlin.js","sources":["wrapper.js","js/arrayUtils.js","js/callableReferenceUtils.js","js/conversions.js","js/core.js","js/long.js","js/markerFunctions.js","js/misc.js","js/polyfills.js","js/rtti.js","runtime/arrayUtils.kt","runtime/Enum.kt","primitiveCompanionObjects.kt","common/src/generated/_Arrays.kt","common/src/generated/_Ranges.kt","unsigned/src/kotlin/UByte.kt","unsigned/src/kotlin/UInt.kt","unsigned/src/kotlin/UShort.kt","src/kotlin/collections/Collections.kt","src/kotlin/collections/Maps.kt","src/kotlin/collections/Sets.kt","src/kotlin/ranges/PrimitiveRanges.kt","src/kotlin/text/StringNumberConversions.kt","src/kotlin/time/Duration.kt","unsigned/src/kotlin/UnsignedUtils.kt","../core/builtins/src/kotlin/internal/InternalAnnotations.kt","src/kotlin/collections/Iterables.kt","src/kotlin/collections/Sequences.kt","src/kotlin/util/Preconditions.kt","js/src/generated/_ArraysJs.kt","src/kotlin/comparisons/Comparisons.kt","src/kotlin/util/Standard.kt","js/src/generated/_ComparisonsJs.kt","unsigned/src/kotlin/ULong.kt","common/src/generated/_Collections.kt","js/src/kotlin/collections.kt","src/kotlin/collections/Iterators.kt","common/src/generated/_Comparisons.kt","common/src/generated/_Maps.kt","common/src/generated/_OneToManyTitlecaseMappings.kt","js/src/kotlin/text/char.kt","js/src/kotlin/text/string.kt","src/kotlin/text/Char.kt","src/kotlin/CharCode.kt","common/src/generated/_Sequences.kt","common/src/generated/_Sets.kt","common/src/generated/_Strings.kt","src/
```

kotlin/text/Strings.kt", "unsigned/src/kotlin/UByteArray.kt", "unsigned/src/kotlin/UIntArray.kt", "unsigned/src/kotlin/ULongArray.kt", "unsigned/src/kotlin/UShortArray.kt", "common/src/generated/_UArrays.kt", "common/src/generated/_UCollections.kt", "common/src/generated/_UComparisons.kt", "common/src/generated/_URanges.kt", "common/src/generated/_USequences.kt", "common/src/kotlin/ExceptionsH.kt", "common/src/kotlin/JsAnnotationsH.kt", "common/src/kotlin/ioH.kt", "builtin-sources/Collections.kt", "builtin-sources/Unit.kt", "builtin-sources/annotation/Annotations.kt", "src/kotlin/builtins.kt", "src/kotlin/jsTypeOf.kt", "src/kotlin/kotlin.kt", "src/kotlin/charCode_js-v1.kt", "src/kotlin/coroutines/CoroutineImpl.kt", "src/kotlin/util/Result.kt", "src/kotlin/coroutines/Continuation.kt", "src/kotlin/coroutines/intrinsics/IntrinsicsJs.kt", "src/kotlin/currentBeMisc.kt", "src/kotlin/exceptions.kt", "src/kotlin/jsOperators.kt", "src/kotlin/math_js-v1.kt", "src/kotlin/numbers_js-v1.kt", "src/kotlin/reflection_js-v1.kt", "src/kotlin/text/numberConversions_js-v1.kt", "js/src/kotlin/js.arrays/fill.kt", "js/src/kotlin/js.arrays/sort.kt", "js/src/generated/_CharCategories.kt", "js/src/generated/_CollectionsJs.kt", "js/src/generated/_DigitChars.kt", "js/src/generated/_LetterChars.kt", "js/src/generated/_OtherLowercaseChars.kt", "js/src/generated/_OtherUppercaseChars.kt", "js/src/generated/_StringsJs.kt", "js/src/generated/_TitlecaseMappings.kt", "js/src/generated/_UArraysJs.kt", "js/src/generated/_WhitespaceChars.kt", "js/src/kotlin/Comparator.kt", "js/src/kotlin/annotations.kt", "js/src/kotlin/annotationsJVM.kt", "js/src/kotlin/collections/AbstractMutableCollection.kt", "js/src/kotlin/collections/AbstractMutableList.kt", "js/src/kotlin/collections/AbstractMutableMap.kt", "js/src/kotlin/collections/AbstractMutableSet.kt", "js/src/kotlin/collections/ArrayList.kt", "js/src/kotlin/collections/ArraySorting.kt", "js/src/kotlin/collections/ArraysJs.kt", "js/src/kotlin/collections/EqualityComparator.kt", "js/src/kotlin/collections/HashMap.kt", "js/src/kotlin/collections/HashSet.kt", "js/src/kotlin/collections/InternalHashCodeMap.kt", "js/src/kotlin/collections/InternalMap.kt", "js/src/kotlin/collections/InternalStringMap.kt", "js/src/kotlin/collections/LinkedHashMap.kt", "js/src/kotlin/collections/LinkedHashSet.kt", "js/src/kotlin/concurrent.kt", "js/src/kotlin/console.kt", "js/src/kotlin/coroutines/SafeContinuationJs.kt", "js/src/kotlin/coroutines/cancellation/CancellationException.kt", "js/src/kotlin/coroutines/js/internal/EmptyContinuation.kt", "js/src/kotlin/date.kt", "js/src/kotlin/dom/Builders.kt", "js/src/kotlin/dom/Classes.kt", "js/src/kotlin/dom/Dom.kt", "js/src/kotlin/dom/EventListener.kt", "js/src/kotlin/dom/ItemArrayLike.kt", "js/src/kotlin/dom/Mutations.kt", "js/src/kotlin/dynamic.kt", "js/src/kotlin/exceptionUtils.kt", "js/src/kotlin/grouping.kt", "src/kotlin/collections/Grouping.kt", "js/src/kotlin/internalAnnotations.kt", "js/src/kotlin/json.kt", "js/src/kotlin/math.kt", "js/src/kotlin/numbers.kt", "js/src/kotlin/promise.kt", "js/src/kotlin/random/PlatformRandom.kt", "js/src/kotlin/reflect/AssociatedObjects.kt", "js/src/kotlin/reflect/JsClass.kt", "js/src/kotlin/reflect/KClassImpl.kt", "js/src/kotlin/reflect/KClassesImpl.kt", "js/src/kotlin/reflect/KTypeHelpers.kt", "js/src/kotlin/reflect/KTypeImpl.kt", "js/src/kotlin/reflect/KTypeParameterImpl.kt", "js/src/kotlin/reflect/primitives.kt", "js/src/kotlin/reflect/reflection.kt", "js/src/kotlin/regex.kt", "js/src/kotlin/sequence.kt", "js/src/kotlin/text/CharCategoryJS.kt", "js/src/kotlin/text/CharacterCodingExceptionJs.kt", "js/src/kotlin/text/StringBuilderJs.kt", "js/src/kotlin/text/numberConversions.kt", "js/src/kotlin/text/regex.kt", "src/kotlin/text/StringBuilder.kt", "js/src/kotlin/text/stringsCode.kt", "js/src/kotlin/text/utf8Encoding.kt", "js/src/kotlin/throwableExtensions.kt", "js/src/kotlin/time/DurationJs.kt", "js/src/kotlin/time/DurationUnit.kt", "js/src/kotlin/time/MonoTimeSource.kt", "js/src/kotlinx/dom/Builders.kt", "js/src/kotlinx/dom/Classes.kt", "src/kotlin/text/regex/RegexExtensions.kt", "js/src/kotlinx/dom/Dom.kt", "js/src/kotlinx/dom/Mutations.kt", "js/src/org.w3c/deprecated.kt", "js/src/org.w3c/org.khronos.webgl.kt", "js/src/org.w3c/org.w3c.dom.clipboard.kt", "js/src/org.w3c/org.w3c.dom.css.kt", "js/src/org.w3c/org.w3c.dom.encryptedmedia.kt", "js/src/org.w3c/org.w3c.dom.events.kt", "js/src/org.w3c/org.w3c.dom.kt", "js/src/org.w3c/org.w3c.fetch.kt", "js/src/org.w3c/org.w3c.dom.mediacapture.kt", "js/src/org.w3c/org.w3c.dom.media.source.kt", "js/src/org.w3c/org.w3c.dom.pointerevents.kt", "js/src/org.w3c/org.w3c.dom.svg.kt", "js/src/org.w3c/org.w3c.files.kt", "js/src/org.w3c/org.w3c.notifications.kt", "js/src/org.w3c/org.w3c.workers.kt", "js/src/org.w3c/org.w3c.xhr.kt", "src/kotlin/annotations/Experimental.kt", "src/kotlin/annotations/ExperimentalStdlibApi.kt", "src/kotlin/annotations/Inference.kt", "src/kotlin/annotations/Multiplatform.kt", "src/kotlin/annotations/OptIn.kt", "src/kotlin/collections/AbstractCollection.kt", "src/kotlin/collections/AbstractIterator.kt", "src/kotlin/collections/AbstractList.kt", "src/kotlin/collections/AbstractMap.kt", "src/kotlin/collections/AbstractSet.kt", "src/kotlin/collections/ArrayDeque.kt", "src/kotlin/collections/Arrays.kt", "src/kotlin/collections/BrittleContainsOptimization.kt", "src/kotlin/collections/IndexedValue.kt", "src/kotlin/collections/MapAccessors.kt", "src/kotlin/collections/MapWithDefault.kt", "src/kotlin/collections/Mut

ableCollections.kt", "src/kotlin/collections/PrimitiveIterators.kt", "src/kotlin/collections/ReversedViews.kt", "src/kotlin/collections/SequenceBuilder.kt", "src/kotlin/collections/SlidingWindow.kt", "src/kotlin/collections/UArraySorting.kt", "src/kotlin/comparisons/compareTo.kt", "src/kotlin/contracts/ContractBuilder.kt", "src/kotlin/coroutines/ContinuationInterceptor.kt", "src/kotlin/coroutines/CoroutineContext.kt", "src/kotlin/coroutines/CoroutineContextImpl.kt", "src/kotlin/coroutines/intrinsics/Intrinsics.kt", "src/kotlin/experimental/bitwiseOperations.kt", "src/kotlin/experimental/inferenceMarker.kt", "src/kotlin/internal/Annotations.kt", "src/kotlin/internal/progressionUtil.kt", "src/kotlin/properties/Delegates.kt", "src/kotlin/properties/Interfaces.kt", "src/kotlin/properties/ObservableProperty.kt", "src/kotlin/properties/PropertyReferenceDelegates.kt", "src/kotlin/random/Random.kt", "src/kotlin/random/URandom.kt", "src/kotlin/random/XorWowRandom.kt", "src/kotlin/ranges/ProgressionIterators.kt", "src/kotlin/ranges/Progressions.kt", "src/kotlin/ranges/Range.kt", "src/kotlin/ranges/Ranges.kt", "src/kotlin/reflect/KClasses.kt", "src/kotlin/reflect/KTypeProjection.kt", "src/kotlin/reflect/KVariance.kt", "src/kotlin/reflect/typeOf.kt", "src/kotlin/text/Appendable.kt", "src/kotlin/text/Indent.kt", "src/kotlin/text/Typography.kt", "src/kotlin/text/regex/MatchResult.kt", "src/kotlin/time/DurationUnit.kt", "src/kotlin/time/ExperimentalTime.kt", "src/kotlin/time/TimeSource.kt", "src/kotlin/time/TimeSources.kt", "src/kotlin/time/longSaturatedMath.kt", "src/kotlin/time/measureTime.kt", "src/kotlin/util/DeepRecursive.kt", "src/kotlin/util/FloorDivMod.kt", "src/kotlin/util/HashCode.kt", "src/kotlin/util/KotlinVersion.kt", "src/kotlin/util/Lateinit.kt", "src/kotlin/util/Lazy.kt", "src/kotlin/util/Numbers.kt", "src/kotlin/util/Suspend.kt", "src/kotlin/util/Tuples.kt", "unsigned/src/kotlin/UIntRange.kt", "unsigned/src/kotlin/ULongRange.kt", "unsigned/src/kotlin/UMath.kt", "unsigned/src/kotlin/UNumbers.kt", "unsigned/src/kotlin/UProgressionUtil.kt", "unsigned/src/kotlin/UStrings.kt", "unsigned/src/kotlin/annotations/Unsigned.kt", "common/src/kotlin/MathH.kt", "js/src/kotlin/js/js.math.kt"], "sourcesContent": ["(function (root, factory) {\n if (typeof define === 'function' && define.amd) {\n define('kotlin', ['exports'], factory);\n }\n else if (typeof exports === 'object') {\n factory(module.exports);\n }\n else {\n root.kotlin = {};\n factory(root.kotlin);\n }\n})(this, function (Kotlin) {\n var _ = Kotlin;\n\n insertContent();\n});\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n\nKotlin.isBooleanArray = function (a) {\n return (Array.isArray(a) || a instanceof Int8Array) && a.\$type\$ === \"BooleanArray\";\n};\n\nKotlin.isByteArray = function (a) {\n return a instanceof Int8Array && a.\$type\$!== \"BooleanArray\";\n};\n\nKotlin.isShortArray = function (a) {\n return a instanceof Int16Array;\n};\n\nKotlin.isCharArray = function (a) {\n return a instanceof Uint16Array && a.\$type\$ === \"CharArray\";\n};\n\nKotlin.isIntArray = function (a) {\n return a instanceof Int32Array;\n};\n\nKotlin.isFloatArray = function (a) {\n return a instanceof Float32Array;\n};\n\nKotlin.isDoubleArray = function (a) {\n return a instanceof Float64Array;\n};\n\nKotlin.isLongArray = function (a) {\n return Array.isArray(a) && a.\$type\$ === \"LongArray\";\n};\n\nKotlin.isArray = function (a) {\n return Array.isArray(a) && !a.\$type\$;\n};\n\nKotlin.isArrayish = function (a) {\n return Array.isArray(a) || ArrayBuffer.isView(a);\n};\n\nKotlin.arrayToString = function (a) {\n if (a === null) return \"null\";\n var toString = Kotlin.isCharArray(a) ? String.fromCharCode : Kotlin.toString;\n return "[" + Array.prototype.map.call(a, function(e) { return toString(e); }).join(", ") + "]";\n};\n\nKotlin.arrayDeepToString = function (arr) {\n return Kotlin.kotlin.collections.contentDeepToStringImpl(arr);\n};\n\nKotlin.arrayEquals = function (a, b) {\n if (a === b) {\n return true;\n }\n if (a === null || b === null || !Kotlin.isArrayish(b) || a.length !== b.length) {\n return false;\n }\n\n for (var i = 0, n = a.length; i < n; i++) {\n if (!Kotlin.equals(a[i], b[i])) {\n return false;\n }\n }\n return true;\n};\n\nKotlin.arrayDeepEquals = function (a, b) {\n return Kotlin.kotlin.collections.contentDeepEqualsImpl(a, b);\n};\n\nKotlin.arrayHashCode = function (arr) {\n if (arr === null) return 0;\n var result = 1;\n for (var i = 0, n = arr.length; i < n; i++) {\n result = ((31 * result | 0) + Kotlin.hashCode(arr[i])) | 0;\n }\n return result;\n};\n\nKotlin.arrayDeepHashCode = function (arr) {\n return Kotlin.kotlin.collections.contentDeepHashCodeImpl(arr);\n};\n\nKotlin.primitiveArraySort = function (array) {\n array.sort(Kotlin.doubleCompareTo);\n};\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the

```

license/LICENSE.txt file.\n */\n\nKotlin.getCallableRef = function(name, f) {\n  f.callableName = name;\n  return
f;\n};\n\nKotlin.getPropertyCallableRef = function(name, paramCount, getter, setter) {\n  getter.get = getter;\n
getter.set = setter;\n  getter.callableName = name;\n  return getPropertyRefClass(getter, setter,
propertyRefClassMetadataCache[paramCount]);\n};\n\nfunction getPropertyRefClass(obj, setter, cache) {\n
obj.$metadata$ = getPropertyRefMetadata(typeof setter === \"function\" ? cache.mutable : cache.immutable);\n
obj.constructor = obj;\n  return obj;\n}\n\nvar propertyRefClassMetadataCache = [\n  {\n    mutable: { value:
null, implementedInterface: function () {\n      return Kotlin.kotlin.reflect.KMutableProperty0 }\n    },\n
immutable: { value: null, implementedInterface: function () {\n      return Kotlin.kotlin.reflect.KProperty0 }\n
  }\n  },\n  {\n    mutable: { value: null, implementedInterface: function () {\n      return
Kotlin.kotlin.reflect.KMutableProperty1 }\n    },\n    immutable: { value: null, implementedInterface: function
() {\n      return Kotlin.kotlin.reflect.KProperty1 }\n    }\n  }\n];\n\nfunction getPropertyRefMetadata(cache)
{\n  if (cache.value === null) {\n    cache.value = {\n      interfaces: [cache.implementedInterface()],\n
baseClass: null,\n      functions: {},\n      properties: {},\n      types: {},\n      staticMembers: {}\n
};\n  }\n  return cache.value;\n}\n\n\"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\nKotlin.toShort = function (a) {\n  return (a & 0xFFFF) << 16 >>
16;\n};\n\nKotlin.toByte = function (a) {\n  return (a & 0xFF) << 24 >> 24;\n};\n\nKotlin.toChar = function (a) {\n
return a & 0xFFFF;\n};\n\nKotlin.numberToLong = function (a) {\n  return a instanceof Kotlin.Long ? a :
Kotlin.Long.fromNumber(a);\n};\n\nKotlin.numberToInt = function (a) {\n  return a instanceof Kotlin.Long ?
a.toInt() : Kotlin.doubleToInt(a);\n};\n\nKotlin.numberToShort = function (a) {\n  return
Kotlin.toShort(Kotlin.numberToInt(a));\n};\n\nKotlin.numberToByte = function (a) {\n  return
Kotlin.toByte(Kotlin.numberToInt(a));\n};\n\nKotlin.numberToDouble = function (a) {\n  return
+a;\n};\n\nKotlin.numberToChar = function (a) {\n  return
Kotlin.toChar(Kotlin.numberToInt(a));\n};\n\nKotlin.doubleToInt = function(a) {\n  if (a > 2147483647) return
2147483647;\n  if (a < -2147483648) return -2147483648;\n  return a | 0;\n};\n\nKotlin.toBoxedChar = function
(a) {\n  if (a == null) return a;\n  if (a instanceof Kotlin.BoxedChar) return a;\n  return new
Kotlin.BoxedChar(a);\n};\n\nKotlin.unboxChar = function(a) {\n  if (a == null) return a;\n  return
Kotlin.toChar(a);\n};\n\n\"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\nKotlin.equals = function (obj1, obj2) {\n  if (obj1 == null) {\n    return obj2 ==
null;\n  }\n  if (obj2 == null) {\n    return false;\n  }\n  if (obj1 !== obj2) {\n    return obj2 !== obj2;\n
}\n  if (typeof obj1 === \"object\" && typeof obj1.equals === \"function\") {\n    return obj1.equals(obj2);\n
}\n  if (typeof obj1 === \"number\" && typeof obj2 === \"number\") {\n    return obj1 === obj2 && (obj1 !==
0 || 1 / obj1 === 1 / obj2);\n  }\n  return obj1 === obj2;\n};\n\nKotlin.hashCode = function (obj) {\n  if (obj ==
null) {\n    return 0;\n  }\n  var objType = typeof obj;\n  if (\"object\" === objType) {\n    return \"function\"
=== typeof obj.hashCode ? obj.hashCode() : getObjectHashCode(obj);\n  }\n  if (\"function\" === objType) {\n
return getObjectHashCode(obj);\n  }\n  if (\"number\" === objType) {\n    return
Kotlin.numberHashCode(obj);\n  }\n  if (\"boolean\" === objType) {\n    return Number(obj)\n  }\n  var str
= String(obj);\n  return getStringHashCode(str);\n};\n\nKotlin.toString = function (o) {\n  if (o == null) {\n
return \"null\";\n  }\n  else if (Kotlin.isArrayish(o)) {\n    return \"[...]\";\n  }\n  else {\n    return
o.toString();\n  }\n};\n\n/** @const */\nvar POW_2_32 = 4294967296;\n// TODO: consider switching to Symbol
type once we are on ES6.\n/** @const */\nvar OBJECT_HASH_CODE_PROPERTY_NAME =
\"kotlinHashCodeValue$\";\n\nfunction getObjectHashCode(obj) {\n  if
(! (OBJECT_HASH_CODE_PROPERTY_NAME in obj)) {\n    var hash = (Math.random() * POW_2_32) | 0; //
Make 32-bit signed integer.\n    Object.defineProperty(obj, OBJECT_HASH_CODE_PROPERTY_NAME, {\n
value: hash, enumerable: false });\n  }\n  return
obj[OBJECT_HASH_CODE_PROPERTY_NAME];\n}\n\nfunction getStringHashCode(str) {\n  var hash = 0;\n
for (var i = 0; i < str.length; i++) {\n    var code = str.charCodeAt(i);\n    hash = (hash * 31 + code) | 0; // Keep

```

```

it 32-bit.\n } \n return hash;\n}\n\nKotlin.identityHashCode = getObjectHashCode;\n"/\n * Copyright 2010-
2018 JetBrains s.r.o. and Kotlin Programming Language contributors. \n * Use of this source code is governed by
the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// Copyright 2009 The Closure
Library Authors. All Rights Reserved.\n/\n// Licensed under the Apache License, Version 2.0 (the "License");\n//
you may not use this file except in compliance with the License.\n// You may obtain a copy of the License at\n/\n//
http://www.apache.org/licenses/LICENSE-2.0\n/\n// Unless required by applicable law or agreed to in writing,
software\n// distributed under the License is distributed on an "AS-IS" BASIS,\n// WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, either express or implied.\n\n/**\n * Constructs a 64-bit two's-complement integer,
given its low and high 32-bit\n * values as *signed* integers. See the from* functions below for more\n *
convenient ways of constructing Longs.\n */\n * The internal representation of a long is the two given signed, 32-bit
values.\n * We use 32-bit pieces because these are the size of integers on which\n * Javascript performs bit-
operations. For operations like addition and\n * multiplication, we split each number into 16-bit pieces, which can
easily be\n * multiplied within Javascript's floating-point representation without overflow\n * or change in sign.\n
*\n * In the algorithms below, we frequently reduce the negative case to the\n * positive case by negating the
input(s) and then post-processing the result.\n * Note that we must ALWAYS check specially whether those values
are MIN_VALUE\n * (-2^63) because -MIN_VALUE == MIN_VALUE (since 2^63 cannot be represented as\n * a
positive number, it overflows back into a negative). Not handling this\n * case would often result in infinite
recursion.\n */\n * @param {number} low The low (signed) 32 bits of the long.\n * @param {number} high The
high (signed) 32 bits of the long.\n * @constructor\n * @final\n */\nKotlin.Long = function(low, high) {\n /**\n *
@type {number}\n * @private\n */\n this.low_ = low | 0; // force into 32 signed bits.\n\n /**\n * @type
{number}\n * @private\n */\n this.high_ = high | 0; // force into 32 signed bits.\n};\n\nKotlin.Long.$metadata$ =
{\n kind: "class",\n simpleName: "Long",\n interfaces:[]\n};\n\n// NOTE: Common constant values
ZERO, ONE, NEG_ONE, etc. are defined below the\n// from* methods on which they depend.\n\n/**\n * A cache
of the Long representations of small integer values.\n * @type {!Object}\n * @private\n */\nKotlin.Long.IntCache_
= {};\n\n/**\n * Returns a Long representing the given (32-bit) integer value.\n * @param {number} value The
32-bit integer in question.\n * @return {!Kotlin.Long} The corresponding Long value.\n */\nKotlin.Long.fromInt =
function(value) {\n if (-128 <= value && value < 128) {\n var cachedObj = Kotlin.Long.IntCache_[value];\n if
(cachedObj) {\n return cachedObj;\n }\n }\n\n var obj = new Kotlin.Long(value | 0, value < 0 ? -1 : 0);\n if (-
128 <= value && value < 128) {\n Kotlin.Long.IntCache_[value] = obj;\n }\n return obj;\n};\n\n/**\n *
Converts this number value to `Long`. The fractional part, if any, is rounded down towards zero.\n * Returns
zero if this `Double` value is `NaN`, `Long.MIN_VALUE` if it's less than `Long.MIN_VALUE`,\n * `Long.MAX_VALUE`
if it's bigger than `Long.MAX_VALUE`.\n * @param {number} value The number in
question.\n * @return {!Kotlin.Long} The corresponding Long value.\n */\nKotlin.Long.fromNumber =
function(value) {\n if (isNaN(value)) {\n return Kotlin.Long.ZERO;\n } else if (value <= -
Kotlin.Long.TWO_PWR_63_DBL_) {\n return Kotlin.Long.MIN_VALUE;\n } else if (value + 1 >=
Kotlin.Long.TWO_PWR_63_DBL_) {\n return Kotlin.Long.MAX_VALUE;\n } else if (value < 0) {\n return
Kotlin.Long.fromNumber(-value).negate();\n } else {\n return new Kotlin.Long(\n (value %
Kotlin.Long.TWO_PWR_32_DBL_) | 0,\n (value / Kotlin.Long.TWO_PWR_32_DBL_) | 0);\n
}\n};\n\n/**\n * Returns a Long representing the 64-bit integer that comes by concatenating\n * the given high and
low bits. Each is assumed to use 32 bits.\n * @param {number} lowBits The low 32-bits.\n * @param {number}
highBits The high 32-bits.\n * @return {!Kotlin.Long} The corresponding Long value.\n */\nKotlin.Long.fromBits =
function(lowBits, highBits) {\n return new Kotlin.Long(lowBits, highBits);\n};\n\n/**\n * Returns a Long
representation of the given string, written using the given\n * radix.\n * @param {string} str The textual
representation of the Long.\n * @param {number=} opt_radix The radix in which the text is written.\n * @return
{!Kotlin.Long} The corresponding Long value.\n */\nKotlin.Long.fromString = function(str, opt_radix) {\n if
(str.length == 0) {\n throw Error('number format error: empty string');\n }\n\n var radix = opt_radix || 10;\n if
(radix < 2 || 36 < radix) {\n throw Error('radix out of range: ' + radix);\n }\n\n if (str.charAt(0) == '-') {\n return
Kotlin.Long.fromString(str.substring(1), radix).negate();\n } else if (str.indexOf('-') >= 0) {\n throw Error('number

```

```

format error: interior \"-\" character: ' + str);\n } \n \n // Do several (8) digits each time through the loop, so as to\n //
minimize the calls to the very expensive emulated div.\n var radixToPower =
Kotlin.Long.fromNumber(Math.pow(radix, 8));\n var result = Kotlin.Long.ZERO;\n for (var i = 0; i < str.length;
i += 8) {\n var size = Math.min(8, str.length - i);\n var value = parseInt(str.substring(i, i + size), radix);\n if
(size < 8) {\n var power = Kotlin.Long.fromNumber(Math.pow(radix, size));\n result =
result.multiply(power).add(Kotlin.Long.fromNumber(value));\n } else {\n result =
result.multiply(radixToPower);\n result = result.add(Kotlin.Long.fromNumber(value));\n } \n } \n return
result;\n};\n\n\n// NOTE: the compiler should inline these constant values below and then remove\n// these
variables, so there should be no runtime penalty for these.\n\n\n**\n * Number used repeated below in calculations.
This must appear before the\n * first call to any from* function below.\n * @type {number}\n * @private\n
*\nKotlin.Long.TWO_PWR_16_DBL_ = 1 << 16;\n\n\n**\n * @type {number}\n * @private\n
*\nKotlin.Long.TWO_PWR_24_DBL_ = 1 << 24;\n\n\n**\n * @type {number}\n * @private\n
*\nKotlin.Long.TWO_PWR_32_DBL_ =\n Kotlin.Long.TWO_PWR_16_DBL_ *
Kotlin.Long.TWO_PWR_16_DBL_;\n\n\n**\n * @type {number}\n * @private\n
*\nKotlin.Long.TWO_PWR_31_DBL_ =\n Kotlin.Long.TWO_PWR_32_DBL_ / 2;\n\n\n**\n * @type
{number}\n * @private\n *\nKotlin.Long.TWO_PWR_48_DBL_ =\n Kotlin.Long.TWO_PWR_32_DBL_ *
Kotlin.Long.TWO_PWR_16_DBL_;\n\n\n**\n * @type {number}\n * @private\n
*\nKotlin.Long.TWO_PWR_64_DBL_ =\n Kotlin.Long.TWO_PWR_32_DBL_ *
Kotlin.Long.TWO_PWR_32_DBL_;\n\n\n**\n * @type {number}\n * @private\n
*\nKotlin.Long.TWO_PWR_63_DBL_ =\n Kotlin.Long.TWO_PWR_64_DBL_ / 2;\n\n\n**\n @type
{!Kotlin.Long} *\nKotlin.Long.ZERO = Kotlin.Long.fromInt(0);\n\n\n**\n @type {!Kotlin.Long}
*\nKotlin.Long.ONE = Kotlin.Long.fromInt(1);\n\n\n**\n @type {!Kotlin.Long} *\nKotlin.Long.NEG_ONE =
Kotlin.Long.fromInt(-1);\n\n\n**\n @type {!Kotlin.Long} *\nKotlin.Long.MAX_VALUE =\n
Kotlin.Long.fromBits(0xFFFFFFFF | 0, 0x7FFFFFFF | 0);\n\n\n**\n @type {!Kotlin.Long}
*\nKotlin.Long.MIN_VALUE = Kotlin.Long.fromBits(0, 0x80000000 | 0);\n\n\n**\n * @type {!Kotlin.Long}\n *
@private\n *\nKotlin.Long.TWO_PWR_24_ = Kotlin.Long.fromInt(1 << 24);\n\n\n**\n @return {number} The
value, assuming it is a 32-bit integer. *\nKotlin.Long.prototype.toInt = function() {\n return this.low_;\n};\n\n\n**\n
@return {number} The closest floating-point representation to this value. *\nKotlin.Long.prototype.toNumber =
function() {\n return this.high_ * Kotlin.Long.TWO_PWR_32_DBL_ +\n
this.getLowBitsUnsigned();\n};\n\n\n**\n @return {number} The 32-bit hashCode of this value.
*\nKotlin.Long.prototype.hashCode = function() {\n return this.high_ ^ this.low_;\n};\n\n\n**\n * @param
{number=} opt_radix The radix in which the text should be written.\n * @return {string} The textual representation
of this value.\n * @override\n *\nKotlin.Long.prototype.toString = function(opt_radix) {\n var radix = opt_radix ||
10;\n if (radix < 2 || 36 < radix) {\n throw Error('radix out of range: ' + radix);\n } \n \n if (this.isZero()) {\n
return '0';\n } \n \n if (this.isNegative()) {\n if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {\n // We need to
change the Long value before it can be negated, so we remove\n // the bottom-most digit in this base and then
recurse to do the rest.\n var radixLong = Kotlin.Long.fromNumber(radix);\n var div = this.div(radixLong);\n
var rem = div.multiply(radixLong).subtract(this);\n return div.toString(radix) + rem.toInt().toString(radix);\n }
else {\n return '-' + this.negate().toString(radix);\n } \n } \n \n // Do several (5) digits each time through the loop,
so as to\n // minimize the calls to the very expensive emulated div.\n var radixToPower =
Kotlin.Long.fromNumber(Math.pow(radix, 5));\n var rem = this;\n var result = '';\n while (true) {\n var
remDiv = rem.div(radixToPower);\n var intVal = rem.subtract(remDiv.multiply(radixToPower)).toInt();\n var
digits = intVal.toString(radix);\n rem = remDiv;\n if (rem.isZero()) {\n return digits + result;\n } else {\n
while (digits.length < 5) {\n digits = '0' + digits;\n } \n result = '' + digits + result;\n } \n } \n};\n\n\n**\n
@return {number} The high 32-bits as a signed value. *\nKotlin.Long.prototype.getHighBits = function() {\n
return this.high_;\n};\n\n\n**\n @return {number} The low 32-bits as a signed value.
*\nKotlin.Long.prototype.getLowBits = function() {\n return this.low_;\n};\n\n\n**\n @return {number} The low
32-bits as an unsigned value. *\nKotlin.Long.prototype.getLowBitsUnsigned = function() {\n return (this.low_ >=

```

```

0) ?\n    this.low_ : Kotlin.Long.TWO_PWR_32_DBL_ + this.low_;\n};\n\n/**\n * @return {number} Returns
the number of bits needed to represent the absolute\n *
value of this Long.\n
*\nKotlin.Long.prototype.getNumBitsAbs = function() {\n if (this.isNegative()) {\n if
(this.equalsLong(Kotlin.Long.MIN_VALUE)) {\n return 64;\n } else {\n return
this.negate().getNumBitsAbs();\n }\n } else {\n var val = this.high_ != 0 ? this.high_ : this.low_;\n for (var bit
= 31; bit > 0; bit--) {\n if ((val & (1 << bit)) != 0) {\n break;\n }\n }\n return this.high_ != 0 ? bit + 33
: bit + 1;\n }\n};\n\n/**
@return {boolean} Whether this value is zero. *\nKotlin.Long.prototype.isZero =
function() {\n return this.high_ == 0 && this.low_ == 0;\n};\n\n/**
@return {boolean} Whether this value is
negative. *\nKotlin.Long.prototype.isNegative = function() {\n return this.high_ < 0;\n};\n\n/**
@return
{boolean} Whether this value is odd. *\nKotlin.Long.prototype.isOdd = function() {\n return (this.low_ & 1) ==
1;\n};\n\n/**\n * @param {Kotlin.Long} other Long to compare against.\n * @return {boolean} Whether this
Long equals the other.\n *\nKotlin.Long.prototype.equalsLong = function(other) {\n return (this.high_ ==
other.high_) && (this.low_ == other.low_);\n};\n\n/**\n * @param {Kotlin.Long} other Long to compare
against.\n * @return {boolean} Whether this Long does not equal the other.\n
*\nKotlin.Long.prototype.notEqualsLong = function(other) {\n return (this.high_ != other.high_) || (this.low_ !=
other.low_);\n};\n\n/**\n * @param {Kotlin.Long} other Long to compare against.\n * @return {boolean}
Whether this Long is less than the other.\n *\nKotlin.Long.prototype.lessThan = function(other) {\n return
this.compare(other) < 0;\n};\n\n/**\n * @param {Kotlin.Long} other Long to compare against.\n * @return
{boolean} Whether this Long is less than or equal to the other.\n *\nKotlin.Long.prototype.lessThanOrEqual =
function(other) {\n return this.compare(other) <= 0;\n};\n\n/**\n * @param {Kotlin.Long} other Long to
compare against.\n * @return {boolean} Whether this Long is greater than the other.\n
*\nKotlin.Long.prototype.greaterThan = function(other) {\n return this.compare(other) > 0;\n};\n\n/**\n *
@param {Kotlin.Long} other Long to compare against.\n * @return {boolean} Whether this Long is greater than or
equal to the other.\n *\nKotlin.Long.prototype.greaterThanOrEqual = function(other) {\n return
this.compare(other) >= 0;\n};\n\n/**\n * Compares this Long with the given one.\n * @param {Kotlin.Long}
other Long to compare against.\n * @return {number} 0 if they are the same, 1 if the this is greater, and -1\n *
if the given one is greater.\n *\nKotlin.Long.prototype.compare = function(other) {\n if (this.equalsLong(other)) {\n
return 0;\n }\n\n var thisNeg = this.isNegative();\n var otherNeg = other.isNegative();\n if (thisNeg &&
!otherNeg) {\n return -1;\n }\n if (!thisNeg && otherNeg) {\n return 1;\n }\n\n // at this point, the signs are the
same, so subtraction will not overflow\n if (this.subtract(other).isNegative()) {\n return -1;\n } else {\n return
1;\n }\n};\n\n/**
@return {!Kotlin.Long} The negation of this value. *\nKotlin.Long.prototype.negate =
function() {\n if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {\n return Kotlin.Long.MIN_VALUE;\n } else
{\n return this.not().add(Kotlin.Long.ONE);\n }\n};\n\n/**\n * Returns the sum of this and the given Long.\n *
@param {Kotlin.Long} other Long to add to this one.\n * @return {!Kotlin.Long} The sum of this and the given
Long.\n *\nKotlin.Long.prototype.add = function(other) {\n // Divide each number into 4 chunks of 16 bits, and
then sum the chunks.\n\n var a48 = this.high_ >>> 16;\n var a32 = this.high_ & 0xFFFF;\n var a16 = this.low_
>>> 16;\n var a00 = this.low_ & 0xFFFF;\n var b48 = other.high_ >>> 16;\n var b32 = other.high_ & 0xFFFF;\n
var b16 = other.low_ >>> 16;\n var b00 = other.low_ & 0xFFFF;\n var c48 = 0, c32 = 0, c16 = 0, c00 = 0;\n c00
+= a00 + b00;\n c16 += c00 >>> 16;\n c00 &= 0xFFFF;\n c16 += a16 + b16;\n c32 += c16 >>> 16;\n c16 &=
0xFFFF;\n c32 += a32 + b32;\n c48 += c32 >>> 16;\n c32 &= 0xFFFF;\n c48 += a48 + b48;\n c48 &=
0xFFFF;\n return Kotlin.Long.fromBits((c16 << 16) | c00, (c48 << 16) | c32);\n};\n\n/**\n * Returns the
difference of this and the given Long.\n * @param {Kotlin.Long} other Long to subtract from this.\n * @return
{!Kotlin.Long} The difference of this and the given Long.\n *\nKotlin.Long.prototype.subtract = function(other)
{\n return this.add(other.negate());\n};\n\n/**\n * Returns the product of this and the given long.\n * @param
{Kotlin.Long} other Long to multiply with this.\n * @return {!Kotlin.Long} The product of this and the other.\n
*\nKotlin.Long.prototype.multiply = function(other) {\n if (this.isZero()) {\n return Kotlin.Long.ZERO;\n } else
if (other.isZero()) {\n return Kotlin.Long.ZERO;\n }\n\n if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {\n
return other.isOdd() ? Kotlin.Long.MIN_VALUE : Kotlin.Long.ZERO;\n } else if

```

```

(other.equalsLong(Kotlin.Long.MIN_VALUE)) {\n    return this.isOdd() ? Kotlin.Long.MIN_VALUE :
Kotlin.Long.ZERO;\n };\n\n if (this.isNegative()) {\n    if (other.isNegative()) {\n        return
this.negate().multiply(other.negate());\n    } else {\n        return this.negate().multiply(other.negate());\n    }\n } else if
(other.isNegative()) {\n    return this.multiply(other.negate()).negate();\n }\n\n // If both longs are small, use float
multiplication\n if (this.lessThan(Kotlin.Long.TWO_PWR_24_) &&\n
other.lessThan(Kotlin.Long.TWO_PWR_24_)) {\n    return Kotlin.Long.fromNumber(this.toNumber() *
other.toNumber());\n }\n\n // Divide each long into 4 chunks of 16 bits, and then add up 4x4 products.\n // We can
skip products that would overflow.\n\n var a48 = this.high_ >>> 16;\n var a32 = this.high_ & 0xFFFF;\n var a16 =
this.low_ >>> 16;\n var a00 = this.low_ & 0xFFFF;\n\n var b48 = other.high_ >>> 16;\n var b32 = other.high_ &
0xFFFF;\n var b16 = other.low_ >>> 16;\n var b00 = other.low_ & 0xFFFF;\n\n var c48 = 0, c32 = 0, c16 = 0, c00
= 0;\n c00 += a00 * b00;\n c16 += c00 >>> 16;\n c00 &= 0xFFFF;\n c16 += a16 * b00;\n c32 += c16 >>> 16;\n
c16 &= 0xFFFF;\n c16 += a00 * b16;\n c32 += c16 >>> 16;\n c16 &= 0xFFFF;\n c32 += a32 * b00;\n c48 +=
c32 >>> 16;\n c32 &= 0xFFFF;\n c32 += a16 * b16;\n c48 += c32 >>> 16;\n c32 &= 0xFFFF;\n c32 += a00 *
b32;\n c48 += c32 >>> 16;\n c32 &= 0xFFFF;\n c48 += a48 * b00 + a32 * b16 + a16 * b32 + a00 * b48;\n c48
&= 0xFFFF;\n return Kotlin.Long.fromBits((c16 << 16) | c00, (c48 << 16) | c32);\n};\n\n\n/**\n * Returns this
Long divided by the given one.\n * @param {Kotlin.Long} other Long by which to divide.\n * @return
{!Kotlin.Long} This Long divided by the given one.\n */\nKotlin.Long.prototype.div = function(other) {\n    if
(other.isZero()) {\n        throw Error('division by zero');\n    } else if (this.isZero()) {\n        return Kotlin.Long.ZERO;\n
}\n\n    if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {\n        if (other.equalsLong(Kotlin.Long.ONE) ||\n
other.equalsLong(Kotlin.Long.NEG_ONE)) {\n            return Kotlin.Long.MIN_VALUE; // recall that -MIN_VALUE
== MIN_VALUE\n        } else if (other.equalsLong(Kotlin.Long.MIN_VALUE)) {\n            return Kotlin.Long.ONE;\n
        } else {\n            // At this point, we have |other| >= 2, so |this/other| < |MIN_VALUE|.\n            var halfThis =
this.shiftRight(1);\n            var approx = halfThis.div(other).shiftLeft(1);\n            if
(approx.equalsLong(Kotlin.Long.ZERO)) {\n                return other.isNegative() ? Kotlin.Long.ONE :
Kotlin.Long.NEG_ONE;\n            } else {\n                var rem = this.subtract(other.multiply(approx));\n                var result =
approx.add(rem.div(other));\n                return result;\n            }\n        } else if
(other.equalsLong(Kotlin.Long.MIN_VALUE)) {\n            return Kotlin.Long.ZERO;\n        }\n\n        if (this.isNegative()) {\n
if (other.isNegative()) {\n            return this.negate().div(other.negate());\n        } else {\n            return
this.negate().div(other.negate()).negate();\n        }\n    } else if (other.isNegative()) {\n        return
this.div(other.negate()).negate();\n    }\n\n    // Repeat the following until the remainder is less than other: find a\n //
floating-point that approximates remainder / other *from below*, add this\n // into the result, and subtract it from
the remainder. It is critical that\n // the approximate value is less than or equal to the real value so that the\n //
remainder never becomes negative.\n    var res = Kotlin.Long.ZERO;\n    var rem = this;\n    while
(rem.greaterThanOrEqual(other)) {\n        // Approximate the result of division. This may be a little greater or\n //
smaller than the actual value.\n        var approx = Math.max(1, Math.floor(rem.toNumber() / other.toNumber()));\n\n        // We will tweak the approximate result by changing it in the 48-th digit or\n // the smallest non-fractional digit,
whichever is larger.\n        var log2 = Math.ceil(Math.log(approx) / Math.LN2);\n        var delta = (log2 <= 48) ? 1 :
Math.pow(2, log2 - 48);\n        // Decrease the approximation until it is smaller than the remainder. Note\n // that if
it is too large, the product overflows and is negative.\n        var approxRes = Kotlin.Long.fromNumber(approx);\n        var approxRem = approxRes.multiply(other);\n        while (approxRem.isNegative() || approxRem.greaterThan(rem))
{\n            approx -= delta;\n            approxRes = Kotlin.Long.fromNumber(approx);\n            approxRem =
approxRes.multiply(other);\n        }\n\n        // We know the answer can't be zero... and actually, zero would cause\n //
infinite recursion since we would make no progress.\n        if (approxRes.isZero()) {\n            approxRes =
Kotlin.Long.ONE;\n        }\n\n        res = res.add(approxRes);\n        rem = rem.subtract(approxRem);\n    }\n    return
res;\n};\n\n\n/**\n * Returns this Long modulo the given one.\n * @param {Kotlin.Long} other Long by which to
mod.\n * @return {!Kotlin.Long} This Long modulo the given one.\n */\nKotlin.Long.prototype.modulo =
function(other) {\n    return this.subtract(this.div(other).multiply(other));\n};\n\n\n/**\n * Returns {!Kotlin.Long} The
bitwise-NOT of this value.\n */\nKotlin.Long.prototype.not = function() {\n    return Kotlin.Long.fromBits(~this.low_,

```

```

~this.high_);};\n\n/**\n * Returns the bitwise-AND of this Long and the given one.\n * @param {Kotlin.Long}
other The Long with which to AND.\n * @return {!Kotlin.Long} The bitwise-AND of this and the other.\n
*/\nKotlin.Long.prototype.and = function(other) {\n return Kotlin.Long.fromBits(this.low_ & other.low_,\n
    this.high_ & other.high_);};\n\n/**\n * Returns the bitwise-OR of this Long and the given one.\n *
@param {Kotlin.Long} other The Long with which to OR.\n * @return {!Kotlin.Long} The bitwise-OR of this and
the other.\n */\nKotlin.Long.prototype.or = function(other) {\n return Kotlin.Long.fromBits(this.low_ |
other.low_,\n    this.high_ | other.high_);};\n\n/**\n * Returns the bitwise-XOR of this Long
and the given one.\n * @param {Kotlin.Long} other The Long with which to XOR.\n * @return {!Kotlin.Long}
The bitwise-XOR of this and the other.\n */\nKotlin.Long.prototype.xor = function(other) {\n return
Kotlin.Long.fromBits(this.low_ ^ other.low_,\n    this.high_ ^ other.high_);};\n\n/**\n *
Returns this Long with bits shifted to the left by the given amount.\n * @param {number} numBits The number of
bits by which to shift.\n * @return {!Kotlin.Long} This shifted to the left by the given amount.\n
*/\nKotlin.Long.prototype.shiftLeft = function(numBits) {\n numBits &= 63;\n if (numBits == 0) {\n return
this;\n } else {\n var low = this.low_;\n if (numBits < 32) {\n var high = this.high_;\n return
Kotlin.Long.fromBits(\n low << numBits,\n (high << numBits) | (low >>> (32 - numBits));\n } else
{\n return Kotlin.Long.fromBits(0, low << (numBits - 32));\n }\n };\n\n/**\n * Returns this Long with
bits shifted to the right by the given amount.\n * @param {number} numBits The number of bits by which to shift.\n
* @return {!Kotlin.Long} This shifted to the right by the given amount.\n */\nKotlin.Long.prototype.shiftRight =
function(numBits) {\n numBits &= 63;\n if (numBits == 0) {\n return this;\n } else {\n var high = this.high_;\n
if (numBits < 32) {\n var low = this.low_;\n return Kotlin.Long.fromBits(\n (low >>> numBits) | (high
<<< (32 - numBits)),\n high >> numBits;\n } else {\n return Kotlin.Long.fromBits(\n high >>
(numBits - 32),\n high >= 0 ? 0 : -1);\n }\n };\n\n/**\n * Returns this Long with bits shifted to the right
by the given amount, with\n * zeros placed into the new leading bits.\n * @param {number} numBits The number
of bits by which to shift.\n * @return {!Kotlin.Long} This shifted to the right by the given amount, with\n *
zeros placed into the new leading bits.\n */\nKotlin.Long.prototype.shiftRightUnsigned = function(numBits) {\n numBits
&= 63;\n if (numBits == 0) {\n return this;\n } else {\n var high = this.high_;\n if (numBits < 32) {\n var
low = this.low_;\n return Kotlin.Long.fromBits(\n (low >>> numBits) | (high <<< (32 - numBits)),\n
high >>> numBits;\n } else if (numBits == 32) {\n return Kotlin.Long.fromBits(high, 0);\n } else {\n
return Kotlin.Long.fromBits(high >>> (numBits - 32), 0);\n }\n };\n\n// Support for
Kotlin\nKotlin.Long.prototype.equals = function (other) {\n return other instanceof Kotlin.Long &&
this.equalsLong(other);};\n\nKotlin.Long.prototype.compareTo_11rb$ =
Kotlin.Long.prototype.compare;\n\nKotlin.Long.prototype.inc = function() {\n return
this.add(Kotlin.Long.ONE);};\n\nKotlin.Long.prototype.dec = function() {\n return
this.add(Kotlin.Long.NEG_ONE);};\n\nKotlin.Long.prototype.valueOf = function() {\n return
this.toNumber();};\n\nKotlin.Long.prototype.unaryPlus = function() {\n return
this;};\n\nKotlin.Long.prototype.unaryMinus = Kotlin.Long.prototype.negate;\n\nKotlin.Long.prototype.inv =
Kotlin.Long.prototype.not;\n\nKotlin.Long.prototype.rangeTo = function (other) {\n return new
Kotlin.kotlin.ranges.LongRange(this, other);};\n\n/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\n/**\n * @param {string} id\n * @param {Object} declaration\n
*/\nKotlin.defineModule = function (id, declaration) {\n};\n\nKotlin.defineInlineFunction = function(tag, fun) {\n
return fun;};\n\nKotlin.wrapFunction = function(fun) {\n var f = function() {\n f = fun();\n return
f.apply(this, arguments);};\n return function() {\n return f.apply(this, arguments);};\n
};\n\nKotlin.isTypeOf = function(type) {\n return function (object) {\n return typeof object === type;\n
};\n\nKotlin.isInstanceOf = function (klass) {\n return function (object) {\n return Kotlin.isType(object,
klass);;\n };\n\nKotlin.orNull = function (fn) {\n return function (object) {\n return object == null ||
fn(object);;\n };\n\nKotlin.andPredicate = function (a, b) {\n return function (object) {\n return a(object)
&& b(object);;\n };\n\nKotlin.kotlinModuleMetadata = function (abiVersion, moduleName, data)

```

```

{\n};\n\nKotlin.suspendCall = function(value) {\n  return value;\n};\n\nKotlin.coroutineResult = function(qualifier)
{\n  throwMarkerError();\n};\n\nKotlin.coroutineController = function(qualifier) {\n
throwMarkerError();\n};\n\nKotlin.coroutineReceiver = function(qualifier) {\n
throwMarkerError();\n};\n\nKotlin.setCoroutineResult = function(value, qualifier) {\n
throwMarkerError();\n};\n\nKotlin.getReifiedTypeParameterKType = function(typeParameter) {\n
throwMarkerError();\n};\n\nfunction throwMarkerError() {\n  throw new Error(\n    \"This marker function
should never be called. \" +\n    \"Looks like compiler did not eliminate it properly. \" +\n    \"Please, report
an issue if you caught this exception.\");\n}\n\nKotlin.getFunctionById = function(id, defaultValue) {\n  return
function() {\n    return defaultValue;\n  };\n};\n\n/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\nKotlin.compareTo = function (a, b) {\n  var typeA = typeof a;\n  if
(typeA === \"number\") {\n    if (typeof b === \"number\") {\n      return Kotlin.doubleCompareTo(a, b);\n    }
\n    return Kotlin.primitiveCompareTo(a, b);\n  }\n  if (typeA === \"string\" || typeA === \"boolean\") {\n
return Kotlin.primitiveCompareTo(a, b);\n  }\n  return
a.compareTo_11rb$(b);\n};\n\nKotlin.primitiveCompareTo = function (a, b) {\n  return a < b ? -1 : a > b ? 1 :
0;\n};\n\nKotlin.doubleCompareTo = function (a, b) {\n  if (a < b) return -1;\n  if (a > b) return 1;\n\n  if (a ===
b) {\n    if (a !== 0) return 0;\n\n    var ia = 1 / a;\n    return ia === 1 / b ? 0 : (ia < 0 ? -1 : 1);\n  }\n\n
return a !== a ? (b !== b ? 0 : 1) : -1;\n};\n\nKotlin.charInc = function (value) {\n  return
Kotlin.toChar(value+1);\n};\n\nKotlin.charDec = function (value) {\n  return Kotlin.toChar(value-
1);\n};\n\nKotlin.imul = Math.imul || imul;\n\nKotlin.imulEmulated = imul;\n\nfunction imul(a, b) {\n  return ((a &
0xffff0000) * (b & 0xffff) + (a & 0xffff) * (b | 0)) | 0;\n}\n\n(function() {\n  var buf = new ArrayBuffer(8);\n  var
bufFloat64 = new Float64Array(buf);\n  var bufFloat32 = new Float32Array(buf);\n  var bufInt32 = new
Int32Array(buf);\n  var lowIndex = 0;\n  var highIndex = 1;\n\n  bufFloat64[0] = -1; // bff00000_00000000\n  if
(bufInt32[lowIndex] !== 0) {\n    lowIndex = 1;\n    highIndex = 0;\n  }\n\n  Kotlin.doubleToBits =
function(value) {\n    return Kotlin.doubleToRawBits(isNaN(value) ? NaN : value);\n  };\n\n  Kotlin.doubleToRawBits = function(value) {\n    bufFloat64[0] = value;\n    return
Kotlin.Long.fromBits(bufInt32[lowIndex], bufInt32[highIndex]);\n  };\n\n  Kotlin.doubleFromBits =
function(value) {\n    bufInt32[lowIndex] = value.low_;\n    bufInt32[highIndex] = value.high_;\n    return
bufFloat64[0];\n  };\n\n  Kotlin.floatToBits = function(value) {\n    return Kotlin.floatToRawBits(isNaN(value)
? NaN : value);\n  };\n\n  Kotlin.floatToRawBits = function(value) {\n    bufFloat32[0] = value;\n    return
bufInt32[0];\n  };\n\n  Kotlin.floatFromBits = function(value) {\n    bufInt32[0] = value;\n    return
bufFloat32[0];\n  };\n\n  // returns zero value for number with positive sign bit and non-zero value for number
with negative sign bit.\n  Kotlin.doubleSignBit = function(value) {\n    bufFloat64[0] = value;\n    return
bufInt32[highIndex] & 0x80000000;\n  };\n\n  Kotlin.numberHashCode = function(obj) {\n    if ((obj | 0) ===
obj) {\n      return obj | 0;\n    } else {\n      bufFloat64[0] = obj;\n      return (bufInt32[highIndex]
* 31 | 0) + bufInt32[lowIndex] | 0;\n    }\n  });\n}\n\nKotlin.ensureNotNull = function(x) {\n  return x != null
? x : Kotlin.throwNPE();\n};\n\n/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\nif (typeof String.prototype.startsWith === \"undefined\") {\n
Object.defineProperty(String.prototype, \"startsWith\", {\n  value: function (searchString, position) {\n
position = position || 0;\n    return this.lastIndexOf(searchString, position) === position;\n  }\n});\n\nif
(typeof String.prototype.endsWith === \"undefined\") {\n  Object.defineProperty(String.prototype, \"endsWith\",
{\n  value: function (searchString, position) {\n    var subjectString = this.toString();\n    if (position
=== undefined || position > subjectString.length) {\n      position = subjectString.length;\n    }\n
position -= searchString.length;\n    var lastIndex = subjectString.indexOf(searchString, position);\n
return lastIndex !== -1 && lastIndex === position;\n  }\n});\n}\n\n// ES6 Math polyfills\nif (typeof Math.sign
=== \"undefined\") {\n  Math.sign = function(x) {\n    x = +x; // convert to a number\n    if (x === 0 ||
isNaN(x)) {\n      return Number(x);\n    }\n    return x > 0 ? 1 : -1;\n  };\n}\n\nif (typeof Math.trunc ===

```



```

\undefined\}) {
  Math.trunc = function(x) {
    if (isNaN(x)) {
      return NaN;
    }
    if (x > 0) {
      return Math.floor(x);
    }
    return Math.ceil(x);
  };
}(function() {
  var epsilon = 2.220446049250313E-16;
  var taylor_2_bound = Math.sqrt(epsilon);
  var taylor_n_bound = Math.sqrt(taylor_2_bound);
  var upper_taylor_2_bound = 1/taylor_2_bound;
  var upper_taylor_n_bound = 1/taylor_n_bound;
  if (typeof Math.sinh === \undefined\}) {
    Math.sinh = function(x) {
      if (Math.abs(x) < taylor_n_bound) {
        var result = x;
      }
      if (Math.abs(x) > taylor_2_bound) {
        result += (x * x * x) / 6;
      }
      return result;
    }
  } else {
    var y = Math.exp(x);
    var y1 = 1 / y;
    if (!isFinite(y)) return Math.exp(x - Math.LN2);
    if (!isFinite(y1)) return -Math.exp(-x - Math.LN2);
    return (y - y1) / 2;
  };
}
if (typeof Math.cosh === \undefined\}) {
  Math.cosh = function(x) {
    var y = Math.exp(x);
    var y1 = 1 / y;
    if (!isFinite(y) || !isFinite(y1)) return Math.exp(Math.abs(x) - Math.LN2);
    return (y + y1) / 2;
  };
}
if (typeof Math.tanh === \undefined\}) {
  Math.tanh = function(x) {
    if (Math.abs(x) < taylor_n_bound) {
      var result = x;
    }
    if (Math.abs(x) > taylor_2_bound) {
      result -= (x * x * x) / 3;
    }
    return result;
  }
} else {
  var a = Math.exp(+x), b = Math.exp(-x);
  return a === Infinity ? 1 : b === Infinity ? -1 : (a - b) / (a + b);
}
}
// Inverse hyperbolic function implementations derived from boost special math functions,
// Copyright Eric Ford & Hubert Holin 2001.
if (typeof Math.asinh === \undefined\}) {
  var asinh = function(x) {
    if (x >= +taylor_n_bound) {
      if (x > upper_taylor_n_bound) {
        if (x > upper_taylor_2_bound) {
          // approximation by laurent series in 1/x at 0+ order from -1 to 0
          return Math.log(x) + Math.LN2;
        }
        // approximation by laurent series in 1/x at 0+ order from -1 to 1
        return Math.log(x * 2 + (1 / (x * 2)));
      }
      else if (x <= -taylor_n_bound) {
        return -asinh(-x);
      }
      else {
        // approximation by taylor series in x at 0 up to order 2
        var result = x;
        if (Math.abs(x) >= taylor_2_bound) {
          var x3 = x * x * x;
          // approximation by taylor series in x at 0 up to order 4
          result -= x3 / 6;
        }
        return result;
      }
    }
    Math.asinh = asinh;
  }
}
if (typeof Math.acosh === \undefined\}) {
  Math.acosh = function(x) {
    if (x < 1) {
      return NaN;
    }
    else if (x - 1 >= taylor_n_bound) {
      if (x > upper_taylor_2_bound) {
        // approximation by laurent series in 1/x at 0+ order from -1 to 0
        return Math.log(x) + Math.LN2;
      }
      else {
        return Math.log(x + Math.sqrt(x * x - 1));
      }
    }
    else {
      var y = Math.sqrt(x - 1);
      // approximation by taylor series in y at 0 up to order 2
      var result = y;
      if (y >= taylor_2_bound) {
        var y3 = y * y * y;
        // approximation by taylor series in y at 0 up to order 4
        result -= y3 / 12;
      }
      return Math.sqrt(2) * result;
    }
  };
}
if (typeof Math.atanh === \undefined\}) {
  Math.atanh = function(x) {
    if (Math.abs(x) < taylor_n_bound) {
      var result = x;
    }
    if (Math.abs(x) > taylor_2_bound) {
      result += (x * x * x) / 3;
    }
    return result;
  }
}
return Math.log((1 + x) / (1 - x)) / 2;
}
}
if (typeof Math.log1p === \undefined\}) {
  Math.log1p = function(x) {
    if (Math.abs(x) < taylor_n_bound) {
      var x2 = x * x;
      var x3 = x2 * x;
      var x4 = x3 * x;
      // approximation by taylor series in x at 0 up to order 4
      return (-x4 / 4 + x3 / 3 - x2 / 2 + x);
    }
    return Math.log(x + 1);
  };
}
if (typeof Math.expm1 === \undefined\}) {
  Math.expm1 = function(x) {
    if (Math.abs(x) < taylor_n_bound) {
      var x2 = x * x;
      var x3 = x2 * x;
      var x4 = x3 * x;
      // approximation by taylor series in x at 0 up to order 4
      return (x4 / 24 + x3 / 6 + x2 / 2 + x);
    }
    return Math.exp(x) - 1;
  };
}
}
if (typeof Math.hypot === \undefined\}) {
  Math.hypot = function() {
    var y = 0;
    var length = arguments.length;
    for (var i = 0; i < length; i++) {
      if (arguments[i] === Infinity || arguments[i] === -Infinity) {
        return Infinity;
      }
      y += arguments[i] * arguments[i];
    }
    return Math.sqrt(y);
  };
}
}
if (typeof Math.log10 === \undefined\}) {

```

```

Math.log10 = function(x) {\n    return Math.log(x) * Math.LOG10E;\n    };\n}\nif (typeof Math.log2 ===
\"undefined\") {\n    Math.log2 = function(x) {\n    return Math.log(x) * Math.LOG2E;\n    };\n}\nif (typeof
Math.clz32 === \"undefined\") {\n    Math.clz32 = (function(log, LN2) {\n    return function(x) {\n    var
asUInt = x >>> 0;\n    if (asUInt === 0) {\n    return 32;\n    }\n    return 31 - (log(asUInt) /
LN2 | 0) | 0; // the \"| 0\" acts like math.floor\n    };\n    })(Math.log, Math.LN2);\n}\n\n// For HtmlUnit and
PhantomJs\nif (typeof ArrayBuffer.isView === \"undefined\") {\n    ArrayBuffer.isView = function(a) {\n
return a != null && a.__proto__ != null && a.__proto__.__proto__ === Int8Array.prototype.__proto__;\n
};\n}\n\nif (typeof Array.prototype.fill === \"undefined\") {\n    // Polyfill from https://developer.mozilla.org/en-
US/docs/Web/JavaScript/Reference/Global_Objects/Array/fill#Polyfill\n    Object.defineProperty(Array.prototype,
'fill', {\n    value: function (value) {\n\n    // Steps 1-2.\n    if (this == null) {\n    throw new
TypeError('this is null or not defined');\n    }\n\n    var O = Object(this);\n\n    // Steps 3-5.\n
var len = O.length >>> 0;\n\n    // Steps 6-7.\n    var start = arguments[1];\n    var relativeStart = start
>> 0;\n\n    // Step 8.\n    var k = relativeStart < 0 ?\n    Math.max(len + relativeStart, 0) :\n    Math.min(relativeStart, len);\n\n    // Steps 9-10.\n    var end = arguments[2];\n    var
relativeEnd = end === undefined ?\n    len : end >> 0;\n\n    // Step 11.\n    var finalValue
= relativeEnd < 0 ?\n    Math.max(len + relativeEnd, 0) :\n    Math.min(relativeEnd,
len);\n\n    // Step 12.\n    while (k < finalValue) {\n    O[k] = value;\n    k++;\n    }\n\n    // Step 13.\n    return O;\n    };\n    });\n}\n\n(function() {\n    function normalizeOffset(offset, length)
{\n    if (offset < 0) return Math.max(0, offset + length);\n    return Math.min(offset, length);\n    }\n    function
typedArraySlice(begin, end) {\n    if (typeof end === \"undefined\") {\n    end = this.length;\n    }\n
begin = normalizeOffset(begin || 0, this.length);\n    end = Math.max(begin, normalizeOffset(end, this.length));\n
return new this.constructor(this.subarray(begin, end));\n    }\n\n    var arrays = [Int8Array, Int16Array,
Uint16Array, Int32Array, Float32Array, Float64Array];\n    for (var i = 0; i < arrays.length; ++i) {\n    var
TypedArray = arrays[i];\n    if (typeof TypedArray.prototype.fill === \"undefined\") {\n
Object.defineProperty(TypedArray.prototype, 'fill', {\n    value: Array.prototype.fill\n    });\n    }\n
if (typeof TypedArray.prototype.slice === \"undefined\") {\n    Object.defineProperty(TypedArray.prototype,
'slice', {\n    value: typedArraySlice\n    });\n    }\n    }\n\n    // Patch apply to work with TypedArrays
if needed.\n    try {\n    (function() {}).apply(null, new Int32Array(0))\n    } catch (e) {\n    var apply =
Function.prototype.apply;\n    Object.defineProperty(Function.prototype, 'apply', {\n    value: function(self,
array) {\n    return apply.call(this, self, [].slice.call(array));\n    }\n    });\n    }\n\n    // Patch map to
work with TypedArrays if needed.\n    for (var i = 0; i < arrays.length; ++i) {\n    var TypedArray = arrays[i];\n
if (typeof TypedArray.prototype.map === \"undefined\") {\n    Object.defineProperty(TypedArray.prototype,
'map', {\n    value: function(callback, self) {\n    return [].slice.call(this).map(callback, self);\n
}\n    });\n    }\n    }\n\n    // Patch sort to work with TypedArrays if needed.\n    // TODO: consider to
remove following function and replace it with `Kotlin.doubleCompareTo` (see misc.js)\n    var
totalOrderComparator = function (a, b) {\n    if (a < b) return -1;\n    if (a > b) return 1;\n\n    if (a === b) {\n
if (a !== 0) return 0;\n\n    var ia = 1 / a;\n    return ia === 1 / b ? 0 : (ia < 0 ? -1 : 1);\n    }\n\n
return a !== a ? (b !== b ? 0 : 1) : -1\n    };\n\n    for (var i = 0; i < arrays.length; ++i) {\n    var TypedArray =
arrays[i];\n    if (typeof TypedArray.prototype.sort === \"undefined\") {\n
Object.defineProperty(TypedArray.prototype, 'sort', {\n    value: function(compareFunction) {\n
return Array.prototype.sort.call(this, compareFunction || totalOrderComparator);\n    }\n    });\n    }\n
}\n\n});\n\n\"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. \n * Use
of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
\n *\/\n\nKotlin.Kind = {\n    CLASS: \"class\",\n    INTERFACE: \"interface\",\n    OBJECT:
\"object\"\n};\n\nKotlin.callGetter = function (thisObject, class, propertyName) {\n    var propertyDescriptor =
Object.getOwnPropertyDescriptor(class, propertyName);\n    if (propertyDescriptor != null &&
propertyDescriptor.get != null) {\n    return propertyDescriptor.get.call(thisObject);\n    }\n\n    propertyDescriptor
= Object.getOwnPropertyDescriptor(thisObject, propertyName);\n    if (propertyDescriptor != null && \"value\" in

```

```

propertyDescriptor) {\n    return thisObject[propertyName];\n }\n\n return Kotlin.callGetter(thisObject,
Object.getPrototypeOf(klass), propertyName);\n};\n\nKotlin.callSetter = function (thisObject, class, propertyName,
value) {\n    var propertyDescriptor = Object.getOwnPropertyDescriptor(klass, propertyName);\n    if
(propertyDescriptor != null && propertyDescriptor.set != null) {\n        propertyDescriptor.set.call(thisObject,
value);\n        return;\n    }\n\n    propertyDescriptor = Object.getOwnPropertyDescriptor(thisObject,
propertyName);\n    if (propertyDescriptor != null && \"value\" in propertyDescriptor) {\n
thisObject[propertyName] = value;\n        return\n    }\n\n    Kotlin.callSetter(thisObject,
Object.getPrototypeOf(klass), propertyName, value);\n};\n\nfunction isInheritanceFromInterface(ctor, iface) {\n    if
(ctor === iface) return true;\n\n    var metadata = ctor.$metadata$;\n    if (metadata != null) {\n        var interfaces =
metadata.interfaces;\n        for (var i = 0; i < interfaces.length; i++) {\n            if
(isInheritanceFromInterface(interfaces[i], iface)) {\n                return true;\n            }\n        }\n\n        var
superPrototype = ctor.prototype != null ? Object.getPrototypeOf(ctor.prototype) : null;\n        var superConstructor =
superPrototype != null ? superPrototype.constructor : null;\n        return superConstructor != null &&
isInheritanceFromInterface(superConstructor, iface);\n    }\n\n    /**\n     * @param {*} object\n     * @param
{Function|Object} klass\n     * @returns {Boolean}\n     */\n    Kotlin.isType = function (object, klass) {\n        if (klass ===
Object) {\n            switch (typeof object) {\n                case \"string\":\n                case \"number\":\n                case
\"boolean\":\n                case \"function\":\n                return true;\n                default:\n                return object instanceof
Object;\n            }\n        }\n\n        if (object == null || klass == null || (typeof object !== 'object' && typeof object !==
'function')) {\n            return false;\n        }\n\n        if (typeof klass === \"function\" && object instanceof klass) {\n
return true;\n        }\n\n        var proto = Object.getPrototypeOf(klass);\n        var constructor = proto != null ?
proto.constructor : null;\n        if (constructor != null && \"$metadata$\" in constructor) {\n            var metadata =
constructor.$metadata$;\n            if (metadata.kind === Kotlin.Kind.OBJECT) {\n                return object === klass;\n            }\n        }\n\n        var classMetadata = klass.$metadata$;\n\n        // In WebKit (JavaScriptCore) for some interfaces from
DOM typeof returns \"object\", nevertheless they can be used in RHS of instanceof\n        if (classMetadata == null) {\n            return object instanceof klass;\n        }\n\n        if (classMetadata.kind === Kotlin.Kind.INTERFACE &&
object.constructor != null) {\n            return isInheritanceFromInterface(object.constructor, klass);\n        }\n\n        return
false;\n    };\n\n    Kotlin.isNumber = function (a) {\n        return typeof a === \"number\" || a instanceof
Kotlin.Long;\n    };\n\n    Kotlin.isChar = function (value) {\n        return value instanceof
Kotlin.BoxedChar;\n    };\n\n    Kotlin.isComparable = function (value) {\n        var type = typeof value;\n\n        return type
=== \"string\" ||\n            type === \"boolean\" ||\n            Kotlin.isNumber(value) ||\n            Kotlin.isType(value,
Kotlin.kotlin.Comparable);\n    };\n\n    Kotlin.isCharSequence = function (value) {\n        return typeof value === \"string\"
|| Kotlin.isType(value, Kotlin.kotlin.CharSequence);\n    };
}

/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\n// a package is omitted to get declarations directly under the
module\n\n@PublishedApi\nexternal fun <T> Array(size: Int): Array<T>\n\n@JsName(\"newArray\")\nfun
<T> newArray(size: Int, initialValue: T) = fillArrayVal(Array<T>(size),
initialValue)\n\n@JsName(\"newArrayF\")\ninline fun <T> arrayWithFun(size: Int, init: (Int) -> T) =
fillArrayFun(Array<T>(size), init)\n\n@JsName(\"fillArray\")\ninline fun <T> fillArrayFun(array: Array<T>, init:
(Int) -> T): Array<T> {\n    for (i in 0..array.size - 1) {\n        array[i] = init(i)\n    }\n    return
array\n}\n\n@JsName(\"booleanArray\")\nfun booleanArray(size: Int, init: dynamic): Array<Boolean> {\n    val
result: dynamic = Array<Boolean>(size)\n    result.`$type$` = \"BooleanArray\"\n    return when (init) {\n        null,
true -> fillArrayVal(result, false)\n        false -> result\n        else -> fillArrayFun<Boolean>(result, init)\n    }\n}\n\n@JsName(\"booleanArrayF\")\ninline fun booleanArrayWithFun(size: Int, init: (Int) -> Boolean):
Array<Boolean> = fillArrayFun(booleanArray(size, false),
init)\n\n@JsName(\"charArray\")\n@Suppress(\"UNUSED_PARAMETER\")\nfun charArray(size: Int, init:
dynamic): Array<Char> {\n    val result = js(\"new Uint16Array(size)\")\n    result.`$type$` = \"CharArray\"\n    return when (init) {\n        null, true, false -> result // For consistency\n        else -> fillArrayFun<Char>(result,
init)\n    }\n}\n\n@JsName(\"charArrayF\")\ninline fun charArrayWithFun(size: Int, init: (Int) -> Char):

```

```

Array<Char> {\n    val array = charArray(size, null)\n    for (i in 0..array.size - 1) {\n
@Suppress("UNUSED_VARIABLE") // used in js block\n        val value = init(i)\n        js("array[i] = value;")\n
    }\n    return array\n}\n\n@JsName("untypedCharArrayF")\ninline fun untypedCharArrayWithFun(size: Int, init:
(Int) -> Char): Array<Char> {\n    val array = Array<Char>(size)\n    for (i in 0..array.size - 1) {\n
@Suppress("UNUSED_VARIABLE") // used in js block\n        val value = init(i)\n        js("array[i] = value;")\n
    }\n    return array\n}\n\n@JsName("longArray")\nfun longArray(size: Int, init: dynamic): Array<Long> {\n    val
result: dynamic = Array<Long>(size)\n    result.`$type$` = "LongArray"\n    return when (init) {\n        null, true ->
fillArrayVal(result, 0L)\n        false -> result\n        else -> fillArrayFun<Long>(result, init)\n
    }\n}\n\n@JsName("longArrayF")\ninline fun longArrayWithFun(size: Int, init: (Int) -> Long): Array<Long> =
fillArrayFun(longArray(size, false), init)\n\nprivate fun <T> fillArrayVal(array: Array<T>, initialValue: T): Array<T>
{\n    for (i in 0..array.size - 1) {\n        array[i] = initialValue\n    }\n    return array\n}, /*\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\npublic class
Enum<T : Enum<T>> : Comparable<Enum<T>> {\n    @JsName("name$") private var _name: String = ""\n
@JsName("ordinal$") private var _ordinal: Int = 0\n    val name: String\n        get() = _name\n    val ordinal:
Int\n        get() = _ordinal\n    override fun compareTo(other: Enum<T>) = ordinal.compareTo(other.ordinal)\n
    override fun equals(other: Any?) = this === other\n    override fun hashCode(): Int =
js("Kotlin.identityHashCode")(this)\n    override fun toString() = name\n    companion object {\n        /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.js.internal\n\n@JsName("DoubleCompanionObject")\ninternal object DoubleCompanionObject {\n
@JsName("MIN_VALUE")\n    const val MIN_VALUE: Double = 4.9E-324\n
@JsName("MAX_VALUE")\n    const val MAX_VALUE: Double = 1.7976931348623157E308\n
@JsName("POSITIVE_INFINITY")\n    @Suppress("DIVISION_BY_ZERO")\n    const val
POSITIVE_INFINITY: Double = 1.0 / 0.0\n
@JsName("NEGATIVE_INFINITY")\n    @Suppress("DIVISION_BY_ZERO")\n    const val
NEGATIVE_INFINITY: Double = -1.0 / 0.0\n
@JsName("NaN")\n    @Suppress("DIVISION_BY_ZERO")\n    const val NaN: Double = -(0.0 / 0.0)\n
@JsName("SIZE_BYTES")\n    const val SIZE_BYTES = 8\n
@JsName("SIZE_BITS")\n    const val
SIZE_BITS = 64\n}\n\n@JsName("FloatCompanionObject")\ninternal object FloatCompanionObject {\n
@JsName("MIN_VALUE")\n    const val MIN_VALUE: Float = 1.4E-45F\n
@JsName("MAX_VALUE")\n    const val MAX_VALUE: Float = 3.4028235E38F\n
@JsName("POSITIVE_INFINITY")\n    @Suppress("DIVISION_BY_ZERO")\n    const val
POSITIVE_INFINITY: Float = 1.0F / 0.0F\n
@JsName("NEGATIVE_INFINITY")\n    @Suppress("DIVISION_BY_ZERO")\n    const val
NEGATIVE_INFINITY: Float = -1.0F / 0.0F\n
@JsName("NaN")\n    @Suppress("DIVISION_BY_ZERO")\n    const val NaN: Float = -(0.0F / 0.0F)\n
@JsName("SIZE_BYTES")\n    const val SIZE_BYTES = 4\n
@JsName("SIZE_BITS")\n    const val
SIZE_BITS = 32\n}\n\n@JsName("IntCompanionObject")\ninternal object IntCompanionObject {\n
@JsName("MIN_VALUE")\n    val MIN_VALUE: Int = -2147483647 - 1\n
@JsName("MAX_VALUE")\n    val MAX_VALUE: Int = 2147483647\n
@JsName("SIZE_BYTES")\n    const val SIZE_BYTES = 4\n
@JsName("SIZE_BITS")\n    const val SIZE_BITS = 32\n}\n\n@JsName("LongCompanionObject")\ninternal
object LongCompanionObject {\n    @JsName("MIN_VALUE")\n    val MIN_VALUE: Long =
js("Kotlin.Long.MIN_VALUE")\n
@JsName("MAX_VALUE")\n    val MAX_VALUE: Long =
js("Kotlin.Long.MAX_VALUE")\n
@JsName("SIZE_BYTES")\n    const val SIZE_BYTES = 8\n
@JsName("SIZE_BITS")\n    const val SIZE_BITS = 64\n}\n\n@JsName("ShortCompanionObject")\ninternal
object ShortCompanionObject {\n    @JsName("MIN_VALUE")\n    val MIN_VALUE: Short = -32768\n
@JsName("MAX_VALUE")\n    val MAX_VALUE: Short = 32767\n
@JsName("SIZE_BYTES")\n    const
val SIZE_BYTES = 2\n
@JsName("SIZE_BITS")\n    const val SIZE_BITS =
16\n}\n\n@JsName("ByteCompanionObject")\ninternal object ByteCompanionObject {\n

```

```

@JsName("MIN_VALUE")\n  val MIN_VALUE: Byte = -128\n\n  @JsName("MAX_VALUE")\n  val
MAX_VALUE: Byte = 127\n\n  @JsName("SIZE_BYTES")\n  const val SIZE_BYTES = 1\n\n
@JsName("SIZE_BITS")\n  const val SIZE_BITS = 8\n}\n\n@JsName("CharCompanionObject")\ninternal
object CharCompanionObject {\n  @JsName("MIN_VALUE")\n  public const val MIN_VALUE: Char =
"\u0000"\n\n  @JsName("MAX_VALUE")\n  public const val MAX_VALUE: Char = "\uFFFF"\n\n
@JsName("MIN_HIGH_SURROGATE")\n  public const val MIN_HIGH_SURROGATE: Char = "\uD800"\n\n
@JsName("MAX_HIGH_SURROGATE")\n  public const val MAX_HIGH_SURROGATE: Char =
"\uDBFF"\n\n  @JsName("MIN_LOW_SURROGATE")\n  public const val MIN_LOW_SURROGATE: Char =
"\uDC00"\n\n  @JsName("MAX_LOW_SURROGATE")\n  public const val MAX_LOW_SURROGATE: Char
= "\uDFFF"\n\n  @JsName("MIN_SURROGATE")\n  public const val MIN_SURROGATE: Char =
MIN_HIGH_SURROGATE\n\n  @JsName("MAX_SURROGATE")\n  public const val MAX_SURROGATE:
Char = MAX_LOW_SURROGATE\n\n  @JsName("SIZE_BYTES")\n  const val SIZE_BYTES = 2\n\n
@JsName("SIZE_BITS")\n  const val SIZE_BITS = 16\n}\n\ninternal object StringCompanionObject
{\n}\n\ninternal object BooleanCompanionObject {\n}\n\n", /*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("ArraysKt")\n\npackage
kotlin.collections\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.*\nimport
kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the
size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior
is unspecified.\n *\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> Array<out T>.component1(): T
{\n  return get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1,
throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun ByteArray.component1(): Byte {\n  return
get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun ShortArray.component1(): Short {\n  return
get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component1(): Int {\n  return
get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component1(): Long {\n  return
get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component1(): Float {\n  return
get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component1(): Double {\n  return
get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component1(): Boolean {\n  return
get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component1(): Char {\n  return
get(0)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n

```

```

*\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> Array<out T>.component2(): T {\n    return
get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an
[IndexOutOfBoundsExce]ption except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun ByteArray.component2(): Byte {\n    return
get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an
[IndexOutOfBoundsExce]ption except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun ShortArray.component2(): Short {\n    return
get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an
[IndexOutOfBoundsExce]ption except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component2(): Int {\n    return
get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an
[IndexOutOfBoundsExce]ption except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component2(): Long {\n    return
get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an
[IndexOutOfBoundsExce]ption except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component2(): Float {\n    return
get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an
[IndexOutOfBoundsExce]ption except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component2(): Double {\n    return
get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an
[IndexOutOfBoundsExce]ption except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component2(): Boolean {\n    return
get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an
[IndexOutOfBoundsExce]ption except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component2(): Char {\n    return
get(1)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsExce]ption except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> Array<out T>.component3(): T {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsExce]ption except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun ByteArray.component3(): Byte {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsExce]ption except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun ShortArray.component3(): Short {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsExce]ption except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component3(): Int {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsExce]ption except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component3(): Long {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsExce]ption except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component3(): Float {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsExce]ption except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component3(): Double {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsExce]ption except in Kotlin/JS\n * where the behavior is unspecified.\n

```

```

*\n@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component3(): Boolean {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component3(): Char {\n    return
get(2)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> Array<out T>.component4(): T {\n    return
get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun ByteArray.component4(): Byte {\n    return
get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun ShortArray.component4(): Short {\n    return
get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component4(): Int {\n    return
get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component4(): Long {\n    return
get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component4(): Float {\n    return
get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component4(): Double {\n    return
get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component4(): Boolean {\n    return
get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component4(): Char {\n    return
get(3)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> Array<out T>.component5(): T {\n    return
get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun ByteArray.component5(): Byte {\n    return
get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun ShortArray.component5(): Short {\n    return
get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component5(): Int {\n    return
get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component5(): Long {\n    return
get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n

```

```

*@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component5(): Float {\n return
get(4)\n}\n/**\n * Returns 5th element from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component5(): Double {\n return
get(4)\n}\n/**\n * Returns 5th element from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component5(): Boolean {\n return
get(4)\n}\n/**\n * Returns 5th element from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component5(): Char {\n return
get(4)\n}\n/**\n * Returns `true` if [element] is found in the array.\n *\npublic operator fun
<@kotlin.internal.OnlyInputTypes T> Array<out T>.contains(element: T): Boolean {\n return indexOf(element)
>= 0\n}\n/**\n * Returns `true` if [element] is found in the array.\n *\npublic operator fun
ByteArray.contains(element: Byte): Boolean {\n return indexOf(element) >= 0\n}\n/**\n * Returns `true` if
[element] is found in the array.\n *\npublic operator fun ShortArray.contains(element: Short): Boolean {\n return
indexOf(element) >= 0\n}\n/**\n * Returns `true` if [element] is found in the array.\n *\npublic operator fun
IntArray.contains(element: Int): Boolean {\n return indexOf(element) >= 0\n}\n/**\n * Returns `true` if
[element] is found in the array.\n *\npublic operator fun LongArray.contains(element: Long): Boolean {\n return
indexOf(element) >= 0\n}\n/**\n * Returns `true` if [element] is found in the array.\n *@Deprecated("The
function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'any { it ==
element }' instead to continue using this behavior, or '.asList().contains(element: T)' to get the same search behavior
as in a list.", ReplaceWith("any { it == element }"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.6", hiddenSince = "1.7")\npublic operator fun FloatArray.contains(element: Float): Boolean {\n return any
{ it == element }\n}\n/**\n * Returns `true` if [element] is found in the array.\n *@Deprecated("The function
has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'any { it == element }'
instead to continue using this behavior, or '.asList().contains(element: T)' to get the same search behavior as in a
list.", ReplaceWith("any { it == element }"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.6", hiddenSince = "1.7")\npublic operator fun DoubleArray.contains(element: Double): Boolean {\n return
any { it == element }\n}\n/**\n * Returns `true` if [element] is found in the array.\n *\npublic operator fun
BooleanArray.contains(element: Boolean): Boolean {\n return indexOf(element) >= 0\n}\n/**\n * Returns `true`
if [element] is found in the array.\n *\npublic operator fun CharArray.contains(element: Char): Boolean {\n return
indexOf(element) >= 0\n}\n/**\n * Returns an element at the given [index] or throws an
[IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAt\n *\npublic expect fun <T> Array<out T>.elementAt(index:
Int): T\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index]
is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n *\npublic
expect fun ByteArray.elementAt(index: Int): Byte\n/**\n * Returns an element at the given [index] or throws an
[IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAt\n *\npublic expect fun ShortArray.elementAt(index: Int):
Short\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index]
is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n *\npublic
expect fun IntArray.elementAt(index: Int): Int\n/**\n * Returns an element at the given [index] or throws an
[IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAt\n *\npublic expect fun LongArray.elementAt(index: Int):
Long\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index]
is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n *\npublic
expect fun FloatArray.elementAt(index: Int): Float\n/**\n * Returns an element at the given [index] or throws an
[IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample

```



```

samples.collections.Collections.Elements.elementAt\n *\/\npublic expect fun DoubleArray.elementAt(index: Int):
Double\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the
[index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n
*\/\npublic expect fun BooleanArray.elementAt(index: Int): Boolean\n\n/**\n * Returns an element at the given
[index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAt\n *\/\npublic expect fun CharArray.elementAt(index: Int):
Char\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the
[index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n *\/\n@kotlin.internal.InlineOnly\npublic inline fun
<T> Array<out T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {\n    return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n *\/\n@kotlin.internal.InlineOnly\npublic inline fun
ByteArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Byte): Byte {\n    return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n *\/\n@kotlin.internal.InlineOnly\npublic inline fun
ShortArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Short): Short {\n    return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n *\/\n@kotlin.internal.InlineOnly\npublic inline fun
IntArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Int): Int {\n    return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n *\/\n@kotlin.internal.InlineOnly\npublic inline fun
LongArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Long): Long {\n    return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n *\/\n@kotlin.internal.InlineOnly\npublic inline fun
FloatArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Float): Float {\n    return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n *\/\n@kotlin.internal.InlineOnly\npublic inline fun
DoubleArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Double): Double {\n    return if (index >= 0 &&
index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the
result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n *\/\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Boolean): Boolean {\n    return if (index >= 0 &&
index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the
result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n *\/\n@kotlin.internal.InlineOnly\npublic inline fun
CharArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Char): Char {\n    return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the
[index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n *\/\n@kotlin.internal.InlineOnly\npublic inline fun
<T> Array<out T>.elementAtOrNull(index: Int): T? {\n    return this.getOrNull(index)\n}\n\n/**\n * Returns an
element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n *\/\n@kotlin.internal.InlineOnly\npublic inline fun

```

`ByteArray.elementAtOrNull(index: Int): Byte?` `{\n return this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun`

`ShortArray.elementAtOrNull(index: Int): Short?` `{\n return this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun`

`IntArray.elementAtOrNull(index: Int): Int?` `{\n return this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun`

`LongArray.elementAtOrNull(index: Int): Long?` `{\n return this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun`

`FloatArray.elementAtOrNull(index: Int): Float?` `{\n return this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun`

`DoubleArray.elementAtOrNull(index: Int): Double?` `{\n return this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun`

`BooleanArray.elementAtOrNull(index: Int): Boolean?` `{\n return this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun`

`CharArray.elementAtOrNull(index: Int): Char?` `{\n return this.getOrNull(index)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.find(predicate: (T) -> Boolean): T?` `{\n return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun`

`ByteArray.find(predicate: (Byte) -> Boolean): Byte?` `{\n return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun`

`ShortArray.find(predicate: (Short) -> Boolean): Short?` `{\n return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun`

`IntArray.find(predicate: (Int) -> Boolean): Int?` `{\n return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun`

`LongArray.find(predicate: (Long) -> Boolean): Long?` `{\n return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun`

`FloatArray.find(predicate: (Float) -> Boolean): Float?` `{\n return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun`

`DoubleArray.find(predicate: (Double) -> Boolean): Double?` `{\n return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun`

`BooleanArray.find(predicate: (Boolean) -> Boolean): Boolean?` `{\n return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun`

CharArray.find(predicate: (Char) -> Boolean): Char? {\n return firstOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.findLast(predicate: (T) -> Boolean): T? {\n return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.findLast(predicate: (Byte) -> Boolean): Byte? {\n return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.findLast(predicate: (Short) -> Boolean): Short? {\n return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.findLast(predicate: (Int) -> Boolean): Int? {\n return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.findLast(predicate: (Long) -> Boolean): Long? {\n return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.findLast(predicate: (Float) -> Boolean): Float? {\n return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.findLast(predicate: (Double) -> Boolean): Double? {\n return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.findLast(predicate: (Boolean) -> Boolean): Boolean? {\n return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.findLast(predicate: (Char) -> Boolean): Char? {\n return lastOrNull(predicate)\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun <T> Array<out T>.first(): T {\n if (isEmpty())\n throw NoSuchElementException("Array is empty.")\n return this[0]\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun ByteArray.first(): Byte {\n if (isEmpty())\n throw NoSuchElementException("Array is empty.")\n return this[0]\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun ShortArray.first(): Short {\n if (isEmpty())\n throw NoSuchElementException("Array is empty.")\n return this[0]\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun IntArray.first(): Int {\n if (isEmpty())\n throw NoSuchElementException("Array is empty.")\n return this[0]\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun LongArray.first(): Long {\n if (isEmpty())\n throw NoSuchElementException("Array is empty.")\n return this[0]\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun FloatArray.first(): Float {\n if (isEmpty())\n throw NoSuchElementException("Array is empty.")\n return this[0]\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun DoubleArray.first(): Double {\n if (isEmpty())\n throw NoSuchElementException("Array is empty.")\n return this[0]\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun BooleanArray.first(): Boolean {\n if (isEmpty())\n throw NoSuchElementException("Array is empty.")\n return this[0]\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun CharArray.first(): Char {\n if (isEmpty())\n throw NoSuchElementException("Array is empty.")\n return this[0]\n}\n\n/**\n * Returns the

```

first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is
found.\n *\npublic inline fun <T> Array<out T>.first(predicate: (T) -> Boolean): T {\n    for (element in this) if
(predicate(element)) return element\n    throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws
[NoSuchElementException] if no such element is found.\n *\npublic inline fun ByteArray.first(predicate: (Byte) ->
Boolean): Byte {\n    for (element in this) if (predicate(element)) return element\n    throw
NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the first
element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n
*\npublic inline fun ShortArray.first(predicate: (Short) -> Boolean): Short {\n    for (element in this) if
(predicate(element)) return element\n    throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws
[NoSuchElementException] if no such element is found.\n *\npublic inline fun IntArray.first(predicate: (Int) ->
Boolean): Int {\n    for (element in this) if (predicate(element)) return element\n    throw
NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the first
element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n
*\npublic inline fun LongArray.first(predicate: (Long) -> Boolean): Long {\n    for (element in this) if
(predicate(element)) return element\n    throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws
[NoSuchElementException] if no such element is found.\n *\npublic inline fun FloatArray.first(predicate: (Float) ->
Boolean): Float {\n    for (element in this) if (predicate(element)) return element\n    throw
NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the first
element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n
*\npublic inline fun DoubleArray.first(predicate: (Double) -> Boolean): Double {\n    for (element in this) if
(predicate(element)) return element\n    throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws
[NoSuchElementException] if no such element is found.\n *\npublic inline fun BooleanArray.first(predicate:
(Boolean) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return element\n    throw
NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the first
element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n
*\npublic inline fun CharArray.first(predicate: (Char) -> Boolean): Char {\n    for (element in this) if
(predicate(element)) return element\n    throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the first non-null value produced by [transform] function being applied to
elements of this array in iteration order,\n * or throws [NoSuchElementException] if no non-null value was
produced.\n * \n * @sample samples.collections.Collections.Transformations.firstNotNullOf\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any> Array<out
T>.firstNotNullOf(transform: (T) -> R?): R {\n    return firstNotNullOfOrNull(transform) ?: throw
NoSuchElementException("No element of the array was transformed to a non-null value.")\n}\n\n/**\n * Returns
the first non-null value produced by [transform] function being applied to elements of this array in iteration order,\n
* or `null` if no non-null value was produced.\n * \n * @sample
samples.collections.Collections.Transformations.firstNotNullOf\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any> Array<out
T>.firstNotNullOfOrNull(transform: (T) -> R?): R? {\n    for (element in this) {\n        val result =
transform(element)\n        if (result != null) {\n            return result\n        }\n    }\n    return null\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n *\npublic fun <T> Array<out T>.firstOrNull(): T? {\n    return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n
*\npublic fun ByteArray.firstOrNull(): Byte? {\n    return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the
first element, or `null` if the array is empty.\n *\npublic fun ShortArray.firstOrNull(): Short? {\n    return if
(isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n *\npublic fun

```

IntArray.firstOrNull(): Int? {\n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\npublic fun LongArray.firstOrNull(): Long? {\n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\npublic fun FloatArray.firstOrNull(): Float? {\n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\npublic fun DoubleArray.firstOrNull(): Double? {\n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\npublic fun BooleanArray.firstOrNull(): Boolean? {\n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\npublic fun CharArray.firstOrNull(): Char? {\n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun <T> Array<out T>.firstOrNull(predicate: (T) -> Boolean): T? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun ByteArray.firstOrNull(predicate: (Byte) -> Boolean): Byte? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun ShortArray.firstOrNull(predicate: (Short) -> Boolean): Short? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun IntArray.firstOrNull(predicate: (Int) -> Boolean): Int? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun LongArray.firstOrNull(predicate: (Long) -> Boolean): Long? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun FloatArray.firstOrNull(predicate: (Float) -> Boolean): Float? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun DoubleArray.firstOrNull(predicate: (Double) -> Boolean): Double? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun BooleanArray.firstOrNull(predicate: (Boolean) -> Boolean): Boolean? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun CharArray.firstOrNull(predicate: (Char) -> Boolean): Char? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.getOrNull(index: Int, defaultValue: (Int) -> T): T? {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.getOrNull(index: Int, defaultValue: (Int) -> Byte): Byte? {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.getOrNull(index: Int, defaultValue: (Int) -> Short): Short? {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.getOrNull(index: Int, defaultValue: (Int) -> Int): Int? {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.getOrNull(index: Int, defaultValue: (Int) -> Long): Long? {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n */\n

```

*@\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.getOrElse(index: Int, defaultValue: (Int) -> Float):
Float {\n  return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns
an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of
this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.getOrElse(index: Int, defaultValue:
(Int) -> Double): Double {\n  return if (index >= 0 && index <= lastIndex) get(index) else
defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue]
function if the [index] is out of bounds of this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.getOrElse(index: Int, defaultValue: (Int) -> Boolean): Boolean {\n  return if (index >= 0 && index
<= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result
of calling the [defaultValue] function if the [index] is out of bounds of this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.getOrElse(index: Int, defaultValue: (Int) -> Char):
Char {\n  return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns
an element at the given [index] or `null` if the [index] is out of bounds of this array.\n */\n * @sample
samples.collections.Collections.Elements.getOrElseOrNull\n */\npublic fun <T> Array<out T>.getOrElseOrNull(index: Int): T?
{\n  return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the
given [index] or `null` if the [index] is out of bounds of this array.\n */\n * @sample
samples.collections.Collections.Elements.getOrElseOrNull\n */\npublic fun ByteArray.getOrElseOrNull(index: Int): Byte? {\n
return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given
[index] or `null` if the [index] is out of bounds of this array.\n */\n * @sample
samples.collections.Collections.Elements.getOrElseOrNull\n */\npublic fun ShortArray.getOrElseOrNull(index: Int): Short? {\n
return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given
[index] or `null` if the [index] is out of bounds of this array.\n */\n * @sample
samples.collections.Collections.Elements.getOrElseOrNull\n */\npublic fun IntArray.getOrElseOrNull(index: Int): Int? {\n
return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given
[index] or `null` if the [index] is out of bounds of this array.\n */\n * @sample
samples.collections.Collections.Elements.getOrElseOrNull\n */\npublic fun LongArray.getOrElseOrNull(index: Int): Long? {\n
return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given
[index] or `null` if the [index] is out of bounds of this array.\n */\n * @sample
samples.collections.Collections.Elements.getOrElseOrNull\n */\npublic fun FloatArray.getOrElseOrNull(index: Int): Float? {\n
return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given
[index] or `null` if the [index] is out of bounds of this array.\n */\n * @sample
samples.collections.Collections.Elements.getOrElseOrNull\n */\npublic fun DoubleArray.getOrElseOrNull(index: Int): Double?
{\n  return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the
given [index] or `null` if the [index] is out of bounds of this array.\n */\n * @sample
samples.collections.Collections.Elements.getOrElseOrNull\n */\npublic fun BooleanArray.getOrElseOrNull(index: Int): Boolean?
{\n  return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the
given [index] or `null` if the [index] is out of bounds of this array.\n */\n * @sample
samples.collections.Collections.Elements.getOrElseOrNull\n */\npublic fun CharArray.getOrElseOrNull(index: Int): Char? {\n
return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns first index of [element], or -
1 if the array does not contain element.\n */\n */\npublic fun <@kotlin.internal.OnlyInputTypes T> Array<out
T>.indexOf(element: T): Int {\n  if (element == null) {\n    for (index in indices) {\n      if (this[index] ==
null) {\n        return index\n      }\n    }\n  } else {\n    for (index in indices) {\n      if (element ==
this[index]) {\n        return index\n      }\n    }\n  }\n  return -1\n}\n\n/**\n * Returns first index of
[element], or -1 if the array does not contain element.\n */\n */\npublic fun ByteArray.indexOf(element: Byte): Int {\n
for (index in indices) {\n  if (element == this[index]) {\n    return index\n  }\n}\n  return -
1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\n */\npublic fun
ShortArray.indexOf(element: Short): Int {\n  for (index in indices) {\n    if (element == this[index]) {\n
return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not

```

```

contain element.\n */\npublic fun IntArray.indexOf(element: Int): Int {\n    for (index in indices) {\n        if (element == this[index]) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\npublic fun LongArray.indexOf(element: Long): Int {\n    for (index in indices) {\n        if (element == this[index]) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\n@Deprecated("\nThe function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'indexOfFirst { it == element }' instead to continue using this behavior, or '.asList().indexOf(element: T)' to get the same search behavior as in a list.", ReplaceWith("\nindexOfFirst { it == element }"))\n@DeprecatedSinceKotlin(warningSince = "\n1.4", errorSince = "\n1.6", hiddenSince = "\n1.7")\npublic fun FloatArray.indexOf(element: Float): Int {\n    for (index in indices) {\n        if (element == this[index]) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\n@Deprecated("\nThe function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'indexOfFirst { it == element }' instead to continue using this behavior, or '.asList().indexOf(element: T)' to get the same search behavior as in a list.", ReplaceWith("\nindexOfFirst { it == element }"))\n@DeprecatedSinceKotlin(warningSince = "\n1.4", errorSince = "\n1.6", hiddenSince = "\n1.7")\npublic fun DoubleArray.indexOf(element: Double): Int {\n    for (index in indices) {\n        if (element == this[index]) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\npublic fun BooleanArray.indexOf(element: Boolean): Int {\n    for (index in indices) {\n        if (element == this[index]) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\npublic fun CharArray.indexOf(element: Char): Int {\n    for (index in indices) {\n        if (element == this[index]) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun <T> Array<out T>.indexOfFirst(predicate: (T) -> Boolean): Int {\n    for (index in indices) {\n        if (predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun ByteArray.indexOfFirst(predicate: (Byte) -> Boolean): Int {\n    for (index in indices) {\n        if (predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun ShortArray.indexOfFirst(predicate: (Short) -> Boolean): Int {\n    for (index in indices) {\n        if (predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun IntArray.indexOfFirst(predicate: (Int) -> Boolean): Int {\n    for (index in indices) {\n        if (predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun LongArray.indexOfFirst(predicate: (Long) -> Boolean): Int {\n    for (index in indices) {\n        if (predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun FloatArray.indexOfFirst(predicate: (Float) -> Boolean): Int {\n    for (index in indices) {\n        if (predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun DoubleArray.indexOfFirst(predicate: (Double) -> Boolean): Int {\n    for (index in indices) {\n        if (predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun BooleanArray.indexOfFirst(predicate: (Boolean) -> Boolean): Int {\n    for (index in indices) {\n        if (predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun CharArray.indexOfFirst(predicate: (Char) -> Boolean): Int {\n    for (index in indices) {\n        if (predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the last

```

```

element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun <T>
Array<out T>.indexOfLast(predicate: (T) -> Boolean): Int {\n    for (index in indices.reversed()) {\n        if
(predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
ByteArray.indexOfLast(predicate: (Byte) -> Boolean): Int {\n    for (index in indices.reversed()) {\n        if
(predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
ShortArray.indexOfLast(predicate: (Short) -> Boolean): Int {\n    for (index in indices.reversed()) {\n        if
(predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
IntArray.indexOfLast(predicate: (Int) -> Boolean): Int {\n    for (index in indices.reversed()) {\n        if
(predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
LongArray.indexOfLast(predicate: (Long) -> Boolean): Int {\n    for (index in indices.reversed()) {\n        if
(predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
FloatArray.indexOfLast(predicate: (Float) -> Boolean): Int {\n    for (index in indices.reversed()) {\n        if
(predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
DoubleArray.indexOfLast(predicate: (Double) -> Boolean): Int {\n    for (index in indices.reversed()) {\n        if
(predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
BooleanArray.indexOfLast(predicate: (Boolean) -> Boolean): Int {\n    for (index in indices.reversed()) {\n        if
(predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
CharArray.indexOfLast(predicate: (Char) -> Boolean): Int {\n    for (index in indices.reversed()) {\n        if
(predicate(this[index])) {\n            return index\n        }\n    }\n    return -1\n}\n\n/**\n * Returns the last element.\n */
\n * @throws NoSuchElementException if the array is empty.\n */\n\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun <T> Array<out T>.last(): T {\n    if (isEmpty())\n    throw NoSuchElementException("Array is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last
element.\n */\n\n * @throws NoSuchElementException if the array is empty.\n */\n\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun ByteArray.last(): Byte {\n    if (isEmpty())\n    throw NoSuchElementException("Array is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last
element.\n */\n\n * @throws NoSuchElementException if the array is empty.\n */\n\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun ShortArray.last(): Short {\n    if (isEmpty())\n    throw NoSuchElementException("Array is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last
element.\n */\n\n * @throws NoSuchElementException if the array is empty.\n */\n\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun IntArray.last(): Int {\n    if (isEmpty())\n    throw
NoSuchElementException("Array is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n */
\n * @throws NoSuchElementException if the array is empty.\n */\n\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun LongArray.last(): Long {\n    if (isEmpty())\n    throw
NoSuchElementException("Array is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last
element.\n */\n\n * @throws NoSuchElementException if the array is empty.\n */\n\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun FloatArray.last(): Float {\n    if (isEmpty())\n    throw
NoSuchElementException("Array is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last
element.\n */\n\n * @throws NoSuchElementException if the array is empty.\n */\n\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun DoubleArray.last(): Double {\n    if (isEmpty())\n    throw
NoSuchElementException("Array is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last

```



```

element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic fun BooleanArray.last(): Boolean {\n  if (isEmpty())\n    throw NoSuchElementException("Array is empty.")\n  return this[lastIndex]\n}\n\n/**\n * Returns the last
element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic fun CharArray.last(): Char {\n  if (isEmpty())\n    throw NoSuchElementException("Array is empty.")\n  return this[lastIndex]\n}\n\n/**\n * Returns the last
element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n
* \n * @sample samples.collections.Collections.Elements.last\n *^\npublic inline fun <T> Array<out
T>.last(predicate: (T) -> Boolean): T {\n  for (index in this.indices.reversed()) {\n    val element = this[index]\n    if (predicate(element)) return element\n  }\n  throw NoSuchElementException("Array contains no element
matching the predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic inline fun ByteArray.last(predicate: (Byte) -> Boolean):
Byte {\n  for (index in this.indices.reversed()) {\n    val element = this[index]\n    if (predicate(element)) return
element\n  }\n  throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic inline fun ShortArray.last(predicate: (Short) -> Boolean):
Short {\n  for (index in this.indices.reversed()) {\n    val element = this[index]\n    if (predicate(element))
return element\n  }\n  throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic inline fun IntArray.last(predicate: (Int) -> Boolean): Int
{\n  for (index in this.indices.reversed()) {\n    val element = this[index]\n    if (predicate(element)) return
element\n  }\n  throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic inline fun LongArray.last(predicate: (Long) -> Boolean):
Long {\n  for (index in this.indices.reversed()) {\n    val element = this[index]\n    if (predicate(element))
return element\n  }\n  throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic inline fun FloatArray.last(predicate: (Float) -> Boolean):
Float {\n  for (index in this.indices.reversed()) {\n    val element = this[index]\n    if (predicate(element))
return element\n  }\n  throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic inline fun DoubleArray.last(predicate: (Double) ->
Boolean): Double {\n  for (index in this.indices.reversed()) {\n    val element = this[index]\n    if
(predicate(element)) return element\n  }\n  throw NoSuchElementException("Array contains no element
matching the predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic inline fun BooleanArray.last(predicate: (Boolean) ->
Boolean): Boolean {\n  for (index in this.indices.reversed()) {\n    val element = this[index]\n    if
(predicate(element)) return element\n  }\n  throw NoSuchElementException("Array contains no element
matching the predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n *^\npublic inline fun CharArray.last(predicate: (Char) -> Boolean):

```

```

Char {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element))
            return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/** Returns last index of [element], or -1 if the array does not contain element. */
public fun <@kotlin.internal.OnlyInputTypes T> Array<out T>.lastIndexOf(element: T): Int {
    if (element == null) {
        for (index in indices.reversed()) {
            if (this[index] == null)
                return index
        }
    } else {
        for (index in indices.reversed()) {
            if (element == this[index])
                return index
        }
    }
    return -1
}

/** Returns last index of [element], or -1 if the array does not contain element. */
public fun ByteArray.lastIndexOf(element: Byte): Int {
    for (index in indices.reversed()) {
        if (element == this[index])
            return index
    }
    return -1
}

/** Returns last index of [element], or -1 if the array does not contain element. */
public fun ShortArray.lastIndexOf(element: Short): Int {
    for (index in indices.reversed()) {
        if (element == this[index])
            return index
    }
    return -1
}

/** Returns last index of [element], or -1 if the array does not contain element. */
public fun IntArray.lastIndexOf(element: Int): Int {
    for (index in indices.reversed()) {
        if (element == this[index])
            return index
    }
    return -1
}

/** Returns last index of [element], or -1 if the array does not contain element. */
public fun LongArray.lastIndexOf(element: Long): Int {
    for (index in indices.reversed()) {
        if (element == this[index])
            return index
    }
    return -1
}

/** Returns last index of [element], or -1 if the array does not contain element.
 * @Deprecated("The function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'indexOfLast { it == element }' instead to continue using this behavior, or '.asList().lastIndexOf(element: T)' to get the same search behavior as in a list.", ReplaceWith("`indexOfLast { it == element }`"))
 */
@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.6", hiddenSince = "1.7")
public fun FloatArray.lastIndexOf(element: Float): Int {
    for (index in indices.reversed()) {
        if (element == this[index])
            return index
    }
    return -1
}

/** Returns last index of [element], or -1 if the array does not contain element.
 * @Deprecated("The function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'indexOfLast { it == element }' instead to continue using this behavior, or '.asList().lastIndexOf(element: T)' to get the same search behavior as in a list.", ReplaceWith("`indexOfLast { it == element }`"))
 */
@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.6", hiddenSince = "1.7")
public fun DoubleArray.lastIndexOf(element: Double): Int {
    for (index in indices.reversed()) {
        if (element == this[index])
            return index
    }
    return -1
}

/** Returns last index of [element], or -1 if the array does not contain element. */
public fun BooleanArray.lastIndexOf(element: Boolean): Int {
    for (index in indices.reversed()) {
        if (element == this[index])
            return index
    }
    return -1
}

/** Returns last index of [element], or -1 if the array does not contain element. */
public fun CharArray.lastIndexOf(element: Char): Int {
    for (index in indices.reversed()) {
        if (element == this[index])
            return index
    }
    return -1
}

/** Returns the last element, or `null` if the array is empty. */
public fun <T> Array<out T>.lastOrNull(): T? {
    return if (isEmpty()) null else this[size - 1]
}

/** Returns the last element, or `null` if the array is empty. */
public fun ByteArray.lastOrNull(): Byte? {
    return if (isEmpty()) null else this[size - 1]
}

/** Returns the last element, or `null` if the array is empty. */
public fun ShortArray.lastOrNull(): Short? {
    return if (isEmpty()) null else this[size - 1]
}

/** Returns the last element, or `null` if the array is empty. */
public fun IntArray.lastOrNull(): Int? {
    return if (isEmpty()) null else this[size - 1]
}

/** Returns the last element, or `null` if the array is empty. */
public fun LongArray.lastOrNull(): Long? {
    return if (isEmpty()) null else this[size - 1]
}

/** Returns the last element, or `null` if the array is empty. */
public fun FloatArray.lastOrNull(): Float? {
    return if (isEmpty()) null else this[size - 1]
}

/** Returns the last element, or `null` if the array is empty. */
public fun DoubleArray.lastOrNull(): Double? {
    return if (isEmpty()) null else this[size - 1]
}

```

```

last element, or `null` if the array is empty.
 * \n * @sample samples.collections.Collections.Elements.last
 * \n public fun BooleanArray.lastOrNull(): Boolean? {
 * \n     return if (isEmpty()) null else this[size - 1]
 * \n }
 * \n * Returns the last element, or `null` if the array is empty.
 * \n * @sample
 * \n samples.collections.Collections.Elements.last
 * \n public fun CharArray.lastOrNull(): Char? {
 * \n     return if (isEmpty()) null else this[size - 1]
 * \n }
 * \n * Returns the last element matching the given [predicate], or `null` if
 * \n no such element was found.
 * \n * @sample samples.collections.Collections.Elements.last
 * \n public inline fun <T> Array<out T>.lastOrNull(predicate: (T) -> Boolean): T? {
 * \n     for (index in this.indices.reversed()) {
 * \n         val element = this[index]
 * \n         if (predicate(element)) return element
 * \n     }
 * \n     return null
 * \n }
 * \n * Returns the last element matching the given [predicate], or `null` if no such element was found.
 * \n * @sample
 * \n samples.collections.Collections.Elements.last
 * \n public inline fun ByteArray.lastOrNull(predicate: (Byte) -> Boolean): Byte? {
 * \n     for (index in this.indices.reversed()) {
 * \n         val element = this[index]
 * \n         if (predicate(element)) return element
 * \n     }
 * \n     return null
 * \n }
 * \n * Returns the last element matching the given [predicate], or `null` if no such element was found.
 * \n * @sample
 * \n samples.collections.Collections.Elements.last
 * \n public inline fun ShortArray.lastOrNull(predicate: (Short) -> Boolean): Short? {
 * \n     for (index in this.indices.reversed()) {
 * \n         val element = this[index]
 * \n         if (predicate(element)) return element
 * \n     }
 * \n     return null
 * \n }
 * \n * Returns the last element matching the given [predicate], or `null` if no such element was found.
 * \n * @sample
 * \n samples.collections.Collections.Elements.last
 * \n public inline fun IntArray.lastOrNull(predicate: (Int) -> Boolean): Int? {
 * \n     for (index in this.indices.reversed()) {
 * \n         val element = this[index]
 * \n         if (predicate(element)) return element
 * \n     }
 * \n     return null
 * \n }
 * \n * Returns the last element matching the given [predicate], or `null` if no such element was found.
 * \n * @sample
 * \n samples.collections.Collections.Elements.last
 * \n public inline fun LongArray.lastOrNull(predicate: (Long) -> Boolean): Long? {
 * \n     for (index in this.indices.reversed()) {
 * \n         val element = this[index]
 * \n         if (predicate(element)) return element
 * \n     }
 * \n     return null
 * \n }
 * \n * Returns the last element matching the given [predicate], or `null` if no such element was found.
 * \n * @sample
 * \n samples.collections.Collections.Elements.last
 * \n public inline fun FloatArray.lastOrNull(predicate: (Float) -> Boolean): Float? {
 * \n     for (index in this.indices.reversed()) {
 * \n         val element = this[index]
 * \n         if (predicate(element)) return element
 * \n     }
 * \n     return null
 * \n }
 * \n * Returns the last element matching the given [predicate], or `null` if no such element was found.
 * \n * @sample
 * \n samples.collections.Collections.Elements.last
 * \n public inline fun DoubleArray.lastOrNull(predicate: (Double) -> Boolean): Double? {
 * \n     for (index in this.indices.reversed()) {
 * \n         val element = this[index]
 * \n         if (predicate(element)) return element
 * \n     }
 * \n     return null
 * \n }
 * \n * Returns the last element matching the given [predicate], or `null` if no such element was found.
 * \n * @sample
 * \n samples.collections.Collections.Elements.last
 * \n public inline fun BooleanArray.lastOrNull(predicate: (Boolean) -> Boolean): Boolean? {
 * \n     for (index in this.indices.reversed()) {
 * \n         val element = this[index]
 * \n         if (predicate(element)) return element
 * \n     }
 * \n     return null
 * \n }
 * \n * Returns the last element matching the given [predicate], or `null` if no such element was found.
 * \n * @sample
 * \n samples.collections.Collections.Elements.last
 * \n public inline fun CharArray.lastOrNull(predicate: (Char) -> Boolean): Char? {
 * \n     for (index in this.indices.reversed()) {
 * \n         val element = this[index]
 * \n         if (predicate(element)) return element
 * \n     }
 * \n     return null
 * \n }
 * \n * Returns a random element from this array.
 * \n * @throws NoSuchElementException if this array is empty.
 * \n * @SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.random(): T {
 * \n     return random(Random)
 * \n }
 * \n * Returns a random element from this array.
 * \n * @throws
 * \n NoSuchElementException if this array is empty.
 * \n * @SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.random(): Byte {
 * \n     return random(Random)
 * \n }
 * \n * Returns a random element from
 * \n this array.
 * \n * @throws NoSuchElementException if this array is empty.
 * \n * @SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.random(): Short {
 * \n     return
 * \n random(Random)
 * \n }
 * \n * Returns a random element from this array.
 * \n * @throws

```

```

NoSuchElementException if this array is empty.\n */\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic
inline fun IntArray.random(): Int {\n    return random(Random)\n}\n\n/**\n * Returns a random element from this
array.\n * \n * @throws NoSuchElementException if this array is empty.\n
*/\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.random(): Long {\n    return
random(Random)\n}\n\n/**\n * Returns a random element from this array.\n * \n * @throws
NoSuchElementException if this array is empty.\n */\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic
inline fun FloatArray.random(): Float {\n    return random(Random)\n}\n\n/**\n * Returns a random element from
this array.\n * \n * @throws NoSuchElementException if this array is empty.\n
*/\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.random(): Double {\n    return
random(Random)\n}\n\n/**\n * Returns a random element from this array.\n * \n * @throws
NoSuchElementException if this array is empty.\n */\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic
inline fun BooleanArray.random(): Boolean {\n    return random(Random)\n}\n\n/**\n * Returns a random element
from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n
*/\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.random(): Char {\n    return
random(Random)\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin("1.3")\npublic
fun <T> Array<out T>.random(random: Random): T {\n    if (isEmpty())\n        throw
NoSuchElementException("Array is empty.")\n    return get(random.nextInt(size))\n}\n\n/**\n * Returns a random
element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if
this array is empty.\n */\n@SinceKotlin("1.3")\npublic fun ByteArray.random(random: Random): Byte {\n    if
(isEmpty())\n        throw NoSuchElementException("Array is empty.")\n    return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin("1.3")\npublic
fun ShortArray.random(random: Random): Short {\n    if (isEmpty())\n        throw
NoSuchElementException("Array is empty.")\n    return get(random.nextInt(size))\n}\n\n/**\n * Returns a random
element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if
this array is empty.\n */\n@SinceKotlin("1.3")\npublic fun IntArray.random(random: Random): Int {\n    if
(isEmpty())\n        throw NoSuchElementException("Array is empty.")\n    return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin("1.3")\npublic
fun LongArray.random(random: Random): Long {\n    if (isEmpty())\n        throw
NoSuchElementException("Array is empty.")\n    return get(random.nextInt(size))\n}\n\n/**\n * Returns a random
element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if
this array is empty.\n */\n@SinceKotlin("1.3")\npublic fun FloatArray.random(random: Random): Float {\n    if
(isEmpty())\n        throw NoSuchElementException("Array is empty.")\n    return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin("1.3")\npublic
fun DoubleArray.random(random: Random): Double {\n    if (isEmpty())\n        throw
NoSuchElementException("Array is empty.")\n    return get(random.nextInt(size))\n}\n\n/**\n * Returns a random
element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if
this array is empty.\n */\n@SinceKotlin("1.3")\npublic fun BooleanArray.random(random: Random): Boolean {\n    if
(isEmpty())\n        throw NoSuchElementException("Array is empty.")\n    return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin("1.3")\npublic
fun CharArray.random(random: Random): Char {\n    if (isEmpty())\n        throw NoSuchElementException("Array
is empty.")\n    return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array, or `null` if
this array is empty.\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli

```

```

c inline fun <T> Array<out T>.randomOrNull(): T? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun ByteArray.randomOrNull(): Byte? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a random
element from this array, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun ShortArray.randomOrNull(): Short? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun IntArray.randomOrNull(): Int? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a random
element from this array, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun LongArray.randomOrNull(): Long? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun FloatArray.randomOrNull(): Float? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun DoubleArray.randomOrNull(): Double? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun BooleanArray.randomOrNull(): Boolean? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun CharArray.randomOrNull(): Char? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array using the specified source of randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T> Array<out
T>.randomOrNull(random: Random): T? {\n    if (isEmpty())\n        return null\n    return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
ByteArray.randomOrNull(random: Random): Byte? {\n    if (isEmpty())\n        return null\n    return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
ShortArray.randomOrNull(random: Random): Short? {\n    if (isEmpty())\n        return null\n    return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
IntArray.randomOrNull(random: Random): Int? {\n    if (isEmpty())\n        return null\n    return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
LongArray.randomOrNull(random: Random): Long? {\n    if (isEmpty())\n        return null\n    return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
FloatArray.randomOrNull(random: Random): Float? {\n    if (isEmpty())\n        return null\n    return

```

```

get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
DoubleArray.randomOrNull(random: Random): Double? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
BooleanArray.randomOrNull(random: Random): Boolean? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
CharArray.randomOrNull(random: Random): Char? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or
has more than one element.\n *\npublic fun <T> Array<out T>.single(): T {\n return when (size) {\n 0 ->
throw NoSuchElementException("Array is empty.")\n 1 -> this[0]\n else -> throw
IllegalArgumentException("Array has more than one element.")\n }\n}\n\n/**\n * Returns the single element, or
throws an exception if the array is empty or has more than one element.\n *\npublic fun ByteArray.single(): Byte
{\n return when (size) {\n 0 -> throw NoSuchElementException("Array is empty.")\n 1 -> this[0]\n
else -> throw IllegalArgumentException("Array has more than one element.")\n }\n}\n\n/**\n * Returns the
single element, or throws an exception if the array is empty or has more than one element.\n *\npublic fun
ShortArray.single(): Short {\n return when (size) {\n 0 -> throw NoSuchElementException("Array is
empty.")\n 1 -> this[0]\n else -> throw IllegalArgumentException("Array has more than one element.")\n
}\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or has more than one
element.\n *\npublic fun IntArray.single(): Int {\n return when (size) {\n 0 -> throw
NoSuchElementException("Array is empty.")\n 1 -> this[0]\n else -> throw
IllegalArgumentException("Array has more than one element.")\n }\n}\n\n/**\n * Returns the single element, or
throws an exception if the array is empty or has more than one element.\n *\npublic fun LongArray.single(): Long
{\n return when (size) {\n 0 -> throw NoSuchElementException("Array is empty.")\n 1 -> this[0]\n
else -> throw IllegalArgumentException("Array has more than one element.")\n }\n}\n\n/**\n * Returns the
single element, or throws an exception if the array is empty or has more than one element.\n *\npublic fun
FloatArray.single(): Float {\n return when (size) {\n 0 -> throw NoSuchElementException("Array is
empty.")\n 1 -> this[0]\n else -> throw IllegalArgumentException("Array has more than one element.")\n
}\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or has more than one
element.\n *\npublic fun DoubleArray.single(): Double {\n return when (size) {\n 0 -> throw
NoSuchElementException("Array is empty.")\n 1 -> this[0]\n else -> throw
IllegalArgumentException("Array has more than one element.")\n }\n}\n\n/**\n * Returns the single element, or
throws an exception if the array is empty or has more than one element.\n *\npublic fun BooleanArray.single():
Boolean {\n return when (size) {\n 0 -> throw NoSuchElementException("Array is empty.")\n 1 ->
this[0]\n else -> throw IllegalArgumentException("Array has more than one element.")\n }\n}\n\n/**\n *
Returns the single element, or throws an exception if the array is empty or has more than one element.\n *\npublic
fun CharArray.single(): Char {\n return when (size) {\n 0 -> throw NoSuchElementException("Array is
empty.")\n 1 -> this[0]\n else -> throw IllegalArgumentException("Array has more than one element.")\n
}\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws exception if there is no or
more than one matching element.\n *\npublic inline fun <T> Array<out T>.single(predicate: (T) -> Boolean): T {\n
var single: T? = null\n var found = false\n for (element in this) {\n if (predicate(element)) {\n if
(found) throw IllegalArgumentException("Array contains more than one matching element.")\n single =
element\n found = true\n }\n }\n if (!found) throw NoSuchElementException("Array contains no
element matching the predicate.")\n @Suppress("UNCHECKED_CAST")\n return single as T\n}\n\n/**\n *

```

Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.

```

public inline fun ByteArray.single(predicate: (Byte) -> Boolean): Byte {
    var single: Byte? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching
the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Byte
}

```

* Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.

```

public inline fun ShortArray.single(predicate: (Short) -> Boolean): Short {
    var single: Short? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching
the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Short
}

```

* Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.

```

public inline fun IntArray.single(predicate: (Int) -> Boolean): Int {
    var single: Int? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching
the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Int
}

```

* Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.

```

public inline fun LongArray.single(predicate: (Long) -> Boolean): Long {
    var single: Long? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching
the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Long
}

```

* Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.

```

public inline fun FloatArray.single(predicate: (Float) -> Boolean): Float {
    var single: Float? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching
the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Float
}

```

* Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.

```

public inline fun DoubleArray.single(predicate: (Double) -> Boolean): Double {
    var single: Double? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element
matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Double
}

```

* Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.

```

public inline fun BooleanArray.single(predicate: (Boolean) -> Boolean): Boolean {
    var single: Boolean? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no
element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Boolean
}

```

* Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.

```

public inline fun CharArray.single(predicate: (Char) -> Boolean): Char {
    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no
element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Char
}

```

```

* Returns single element, or `null` if the array is empty or has more than one element.
public fun <T>
Array<out T>.singleOrNull(): T? {
    return if (size == 1) this[0] else null
}

* Returns single element, or `null` if the array is empty or has more than one element.
public fun ByteArray.singleOrNull(): Byte? {
    return if (size == 1) this[0] else null
}

* Returns single element, or `null` if the array is empty or has more than one element.
public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else
    null
}

* Returns single element, or `null` if the array is empty or has more than one element.
public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

* Returns single element, or `null` if the array is empty or has more than one element.
public fun LongArray.singleOrNull(): Long? {
    return if (size == 1) this[0] else null
}

* Returns single element, or `null` if the array is empty or has more than one element.
public fun FloatArray.singleOrNull(): Float? {
    return if (size == 1) this[0] else
    null
}

* Returns single element, or `null` if the array is empty or has more than one element.
public fun DoubleArray.singleOrNull(): Double? {
    return if (size == 1) this[0] else null
}

* Returns single element, or `null` if the array is empty or has more than one element.
public fun BooleanArray.singleOrNull(): Boolean? {
    return if (size == 1) this[0] else null
}

* Returns single element, or `null` if the array is empty or has more than one element.
public fun CharArray.singleOrNull(): Char? {
    return if (size == 1)
    this[0] else null
}

* Returns the single element matching the given [predicate], or `null` if element was not
found or more than one element was found.
public inline fun <T> Array<out T>.singleOrNull(predicate: (T) -
> Boolean): T? {
    var single: T? = null
    var found = false
    for (element in this) {
        if
        (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
        if (!found) return single
    }
}

* Returns the single element matching the given [predicate], or `null` if element was not
found or more than one element was found.
public inline fun
ByteArray.singleOrNull(predicate: (Byte) -> Boolean): Byte? {
    var single: Byte? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
        if (!found) return single
    }
}

* Returns the single element
matching the given [predicate], or `null` if element was not found or more than one element
was found.
public inline fun ShortArray.singleOrNull(predicate: (Short) -> Boolean): Short? {
    var single: Short? = null
    var
    found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single
            = element
            found = true
        }
        if (!found) return single
    }
}

* Returns the
single element matching the given [predicate], or `null` if element was not found or more than
one element was
found.
public inline fun IntArray.singleOrNull(predicate: (Int) -> Boolean): Int? {
    var single: Int? = null
    var
    found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single
            = element
            found = true
        }
        if (!found) return single
    }
}

* Returns the single element matching the given [predicate], or `null` if element was not found or more than one
element was
found.
public inline fun LongArray.singleOrNull(predicate: (Long) -> Boolean): Long? {
    var single: Long? = null
    var
    found = false
    for (element in this) {
        if (predicate(element)) {
            if
            (found) return null
            single = element
            found = true
        }
        if (!found) return single
    }
}

* Returns the single element matching the given [predicate], or `null` if element was not found or
more than one
element was found.
public inline fun FloatArray.singleOrNull(predicate: (Float) -> Boolean):
Float? {
    var single: Float? = null
    var found = false
    for (element in this) {
        if (predicate(element))
        {
            if (found) return null
            single = element
            found = true
        }
        if (!found) return
        null
    }
}

* Returns the single element matching the given [predicate], or `null` if element
was not found or more than one element was found.
public inline fun DoubleArray.singleOrNull(predicate:
(Double) -> Boolean): Double? {
    var single: Double? = null
    var found = false
    for (element in this) {
        if (predicate(element))
        {
            if (found) return null
            single = element
            found = true
        }
        if (!found) return
        null
    }
}

* Returns the single element matching the given
[predicate], or `null` if element was not found or more than one element was found.
public inline fun
BooleanArray.singleOrNull(predicate: (Boolean) -> Boolean): Boolean? {
    var single: Boolean? = null
    var
    found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single
        }
        if (!found) return single
    }
}

```



```

= element\n      found = true\n      }\n      }\n      if (!found) return null\n      return single\n    }\n\n    * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n  * \n\n  public inline fun CharArray.singleOrNull(predicate: (Char) -> Boolean): Char? {\n    var single: Char? = null\n    var found = false\n    for (element in this) {\n      if (predicate(element)) {\n        if (found) return null\n        single = element\n        found = true\n      }\n    }\n    if (!found) return null\n    return single\n  }\n\n  * Returns a list containing all elements except first [n] elements.\n  * \n  * @throws IllegalArgumentException if [n] is negative.\n  * \n  * @sample samples.collections.Collections.Transformations.drop\n\n  public fun <T> Array<out T>.drop(n: Int): List<T> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    return takeLast((size - n).coerceAtLeast(0))\n  }\n\n  * Returns a list containing all elements except first [n] elements.\n  * \n  * @throws IllegalArgumentException if [n] is negative.\n  * \n  * @sample samples.collections.Collections.Transformations.drop\n\n  public fun ByteArray.drop(n: Int): List<Byte> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    return takeLast((size - n).coerceAtLeast(0))\n  }\n\n  * Returns a list containing all elements except first [n] elements.\n  * \n  * @throws IllegalArgumentException if [n] is negative.\n  * \n  * @sample samples.collections.Collections.Transformations.drop\n\n  public fun ShortArray.drop(n: Int): List<Short> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    return takeLast((size - n).coerceAtLeast(0))\n  }\n\n  * Returns a list containing all elements except first [n] elements.\n  * \n  * @throws IllegalArgumentException if [n] is negative.\n  * \n  * @sample samples.collections.Collections.Transformations.drop\n\n  public fun IntArray.drop(n: Int): List<Int> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    return takeLast((size - n).coerceAtLeast(0))\n  }\n\n  * Returns a list containing all elements except first [n] elements.\n  * \n  * @throws IllegalArgumentException if [n] is negative.\n  * \n  * @sample samples.collections.Collections.Transformations.drop\n\n  public fun LongArray.drop(n: Int): List<Long> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    return takeLast((size - n).coerceAtLeast(0))\n  }\n\n  * Returns a list containing all elements except first [n] elements.\n  * \n  * @throws IllegalArgumentException if [n] is negative.\n  * \n  * @sample samples.collections.Collections.Transformations.drop\n\n  public fun FloatArray.drop(n: Int): List<Float> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    return takeLast((size - n).coerceAtLeast(0))\n  }\n\n  * Returns a list containing all elements except first [n] elements.\n  * \n  * @throws IllegalArgumentException if [n] is negative.\n  * \n  * @sample samples.collections.Collections.Transformations.drop\n\n  public fun DoubleArray.drop(n: Int): List<Double> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    return takeLast((size - n).coerceAtLeast(0))\n  }\n\n  * Returns a list containing all elements except first [n] elements.\n  * \n  * @throws IllegalArgumentException if [n] is negative.\n  * \n  * @sample samples.collections.Collections.Transformations.drop\n\n  public fun BooleanArray.drop(n: Int): List<Boolean> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    return takeLast((size - n).coerceAtLeast(0))\n  }\n\n  * Returns a list containing all elements except first [n] elements.\n  * \n  * @throws IllegalArgumentException if [n] is negative.\n  * \n  * @sample samples.collections.Collections.Transformations.drop\n\n  public fun CharArray.drop(n: Int): List<Char> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    return takeLast((size - n).coerceAtLeast(0))\n  }\n\n  * Returns a list containing all elements except last [n] elements.\n  * \n  * @throws IllegalArgumentException if [n] is negative.\n  * \n  * @sample samples.collections.Collections.Transformations.drop\n\n  public fun <T> Array<out T>.dropLast(n: Int): List<T> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    return take((size - n).coerceAtLeast(0))\n  }\n\n  * Returns a list containing all elements except last [n] elements.\n  * \n  * @throws IllegalArgumentException if [n] is negative.\n  * \n  * @sample samples.collections.Collections.Transformations.drop\n\n  public fun ByteArray.dropLast(n: Int): List<Byte> {\n

```

```

require(n >= 0) { \"Requested element count $n is less than zero.\" } \n  return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic fun ShortArray.dropLast(n: Int): List<Short> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" } \n  return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic fun IntArray.dropLast(n: Int): List<Int> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" } \n  return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic fun LongArray.dropLast(n: Int): List<Long> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" } \n  return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic fun FloatArray.dropLast(n: Int): List<Float> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" } \n  return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic fun DoubleArray.dropLast(n: Int): List<Double>
{\n  require(n >= 0) { \"Requested element count $n is less than zero.\" } \n  return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic fun BooleanArray.dropLast(n: Int):
List<Boolean> {\n  require(n >= 0) { \"Requested element count $n is less than zero.\" } \n  return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic fun CharArray.dropLast(n: Int): List<Char> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" } \n  return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given
[predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n *\npublic inline fun <T>
Array<out T>.dropLastWhile(predicate: (T) -> Boolean): List<T> {\n  for (index in lastIndex downTo 0) {\n    if
(!predicate(this[index])) {\n      return take(index + 1)\n    } \n  } \n  return emptyList()\n}\n\n/**\n * Returns
a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic inline fun ByteArray.dropLastWhile(predicate:
(Byte) -> Boolean): List<Byte> {\n  for (index in lastIndex downTo 0) {\n    if (!predicate(this[index])) {\n
return take(index + 1)\n    } \n  } \n  return emptyList()\n}\n\n/**\n * Returns a list containing all elements
except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic inline fun ShortArray.dropLastWhile(predicate:
(Short) -> Boolean): List<Short> {\n  for (index in lastIndex downTo 0) {\n    if (!predicate(this[index])) {\n
return take(index + 1)\n    } \n  } \n  return emptyList()\n}\n\n/**\n * Returns a list containing all elements
except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic inline fun IntArray.dropLastWhile(predicate:
(Int) -> Boolean): List<Int> {\n  for (index in lastIndex downTo 0) {\n    if (!predicate(this[index])) {\n
return take(index + 1)\n    } \n  } \n  return emptyList()\n}\n\n/**\n * Returns a list containing all elements
except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic inline fun LongArray.dropLastWhile(predicate:
(Long) -> Boolean): List<Long> {\n  for (index in lastIndex downTo 0) {\n    if (!predicate(this[index])) {\n

```

```

return take(index + 1)\n    }\n    }\n    return emptyList()\n}\n\n/**\n * Returns a list containing all elements
except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic inline fun FloatArray.dropLastWhile(predicate:
(Float) -> Boolean): List<Float> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n
return take(index + 1)\n        }\n    }\n    return emptyList()\n}\n\n/**\n * Returns a list containing all elements
except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic inline fun DoubleArray.dropLastWhile(predicate:
(Double) -> Boolean): List<Double> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n
return take(index + 1)\n        }\n    }\n    }\n    return emptyList()\n}\n\n/**\n * Returns a list containing all elements
except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic inline fun
BooleanArray.dropLastWhile(predicate: (Boolean) -> Boolean): List<Boolean> {\n    for (index in lastIndex
downTo 0) {\n        if (!predicate(this[index])) {\n            return take(index + 1)\n        }\n    }\n    return
emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given
[predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n *\npublic inline fun
CharArray.dropLastWhile(predicate: (Char) -> Boolean): List<Char> {\n    for (index in lastIndex downTo 0) {\n
if (!predicate(this[index])) {\n        return take(index + 1)\n    }\n    }\n    return emptyList()\n}\n\n/**\n *
Returns a list containing all elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic inline fun <T> Array<out
T>.dropWhile(predicate: (T) -> Boolean): List<T> {\n    var yielding = false\n    val list = ArrayList<T>()\n    for
(item in this)\n        if (yielding)\n            list.add(item)\n        else if (!predicate(item)) {\n            list.add(item)\n
yielding = true\n        }\n    return list\n}\n\n/**\n * Returns a list containing all elements except first elements that
satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n *\npublic
inline fun ByteArray.dropWhile(predicate: (Byte) -> Boolean): List<Byte> {\n    var yielding = false\n    val list =
ArrayList<Byte>()\n    for (item in this)\n        if (yielding)\n            list.add(item)\n        else if (!predicate(item)) {\n
list.add(item)\n            yielding = true\n        }\n    return list\n}\n\n/**\n * Returns a list containing all
elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic inline fun ShortArray.dropWhile(predicate:
(Short) -> Boolean): List<Short> {\n    var yielding = false\n    val list = ArrayList<Short>()\n    for (item in this)\n
if (yielding)\n        list.add(item)\n    else if (!predicate(item)) {\n        list.add(item)\n        yielding =
true\n    }\n    return list\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the
given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n *\npublic inline fun
IntArray.dropWhile(predicate: (Int) -> Boolean): List<Int> {\n    var yielding = false\n    val list =
ArrayList<Int>()\n    for (item in this)\n        if (yielding)\n            list.add(item)\n        else if (!predicate(item)) {\n
list.add(item)\n            yielding = true\n        }\n    return list\n}\n\n/**\n * Returns a list containing all elements
except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic inline fun LongArray.dropWhile(predicate:
(Long) -> Boolean): List<Long> {\n    var yielding = false\n    val list = ArrayList<Long>()\n    for (item in this)\n
if (yielding)\n        list.add(item)\n    else if (!predicate(item)) {\n        list.add(item)\n        yielding =
true\n    }\n    return list\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the
given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n *\npublic inline fun
FloatArray.dropWhile(predicate: (Float) -> Boolean): List<Float> {\n    var yielding = false\n    val list =
ArrayList<Float>()\n    for (item in this)\n        if (yielding)\n            list.add(item)\n        else if (!predicate(item))
{\n            list.add(item)\n            yielding = true\n        }\n    return list\n}\n\n/**\n * Returns a list containing all
elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n *\npublic inline fun DoubleArray.dropWhile(predicate:
(Double) -> Boolean): List<Double> {\n    var yielding = false\n    val list = ArrayList<Double>()\n    for (item in
this)\n        if (yielding)\n            list.add(item)\n        else if (!predicate(item)) {\n            list.add(item)\n
}

```

```

yielding = true\n    }\n    return list\n}\n\n/**\n * Returns a list containing all elements except first elements that
satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic
inline fun BooleanArray.dropWhile(predicate: (Boolean) -> Boolean): List<Boolean> {\n    var yielding = false\n
val list = ArrayList<Boolean>()\n    for (item in this)\n        if (yielding)\n            list.add(item)\n        else if
(!predicate(item)) {\n            list.add(item)\n            yielding = true\n        }\n    return list\n}\n\n/**\n * Returns a list
containing all elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic inline fun CharArray.dropWhile(predicate:
(Char) -> Boolean): List<Char> {\n    var yielding = false\n    val list = ArrayList<Char>()\n    for (item in this)\n
if (yielding)\n        list.add(item)\n    else if (!predicate(item)) {\n        list.add(item)\n        yielding =
true\n    }\n    return list\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n *
@sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun <T> Array<out
T>.filter(predicate: (T) -> Boolean): List<T> {\n    return filterTo(ArrayList<T>(), predicate)\n}\n\n/**\n * Returns
a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n */\npublic inline fun ByteArray.filter(predicate: (Byte) -> Boolean):
List<Byte> {\n    return filterTo(ArrayList<Byte>(), predicate)\n}\n\n/**\n * Returns a list containing only elements
matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline
fun ShortArray.filter(predicate: (Short) -> Boolean): List<Short> {\n    return filterTo(ArrayList<Short>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n */\npublic inline fun IntArray.filter(predicate: (Int) -> Boolean):
List<Int> {\n    return filterTo(ArrayList<Int>(), predicate)\n}\n\n/**\n * Returns a list containing only elements
matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline
fun LongArray.filter(predicate: (Long) -> Boolean): List<Long> {\n    return filterTo(ArrayList<Long>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n */\npublic inline fun FloatArray.filter(predicate: (Float) ->
Boolean): List<Float> {\n    return filterTo(ArrayList<Float>(), predicate)\n}\n\n/**\n * Returns a list containing
only elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n
*/\npublic inline fun DoubleArray.filter(predicate: (Double) -> Boolean): List<Double> {\n    return
filterTo(ArrayList<Double>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given
[predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun
BooleanArray.filter(predicate: (Boolean) -> Boolean): List<Boolean> {\n    return filterTo(ArrayList<Boolean>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n */\npublic inline fun CharArray.filter(predicate: (Char) ->
Boolean): List<Char> {\n    return filterTo(ArrayList<Char>(), predicate)\n}\n\n/**\n * Returns a list containing
only elements matching the given [predicate].\n * \n * @param [predicate] function that takes the index of an element
and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun <T> Array<out
T>.filterIndexed(predicate: (index: Int, T) -> Boolean): List<T> {\n    return filterIndexedTo(ArrayList<T>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @param
[predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate
evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic
inline fun ByteArray.filterIndexed(predicate: (index: Int, Byte) -> Boolean): List<Byte> {\n    return
filterIndexedTo(ArrayList<Byte>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the
given [predicate].\n * \n * @param [predicate] function that takes the index of an element and the element itself\n *
and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun ShortArray.filterIndexed(predicate:
(index: Int, Short) -> Boolean): List<Short> {\n    return filterIndexedTo(ArrayList<Short>(), predicate)\n}\n\n/**\n *
Returns a list containing only elements matching the given [predicate].\n * \n * @param [predicate] function that takes
the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n
*/

```

```

* @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun
IntArray.filterIndexed(predicate: (index: Int, Int) -> Boolean): List<Int> {\n  return
filterIndexedTo(ArrayList<Int>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the
given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and
returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun LongArray.filterIndexed(predicate:
(index: Int, Long) -> Boolean): List<Long> {\n  return filterIndexedTo(ArrayList<Long>(), predicate)\n}\n\n/**\n *
Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes
the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n
* @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun
FloatArray.filterIndexed(predicate: (index: Int, Float) -> Boolean): List<Float> {\n  return
filterIndexedTo(ArrayList<Float>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the
given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and
returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun DoubleArray.filterIndexed(predicate:
(index: Int, Double) -> Boolean): List<Double> {\n  return filterIndexedTo(ArrayList<Double>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param
[predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate
evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic
inline fun BooleanArray.filterIndexed(predicate: (index: Int, Boolean) -> Boolean): List<Boolean> {\n  return
filterIndexedTo(ArrayList<Boolean>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching
the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n *
and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun CharArray.filterIndexed(predicate:
(index: Int, Char) -> Boolean): List<Char> {\n  return filterIndexedTo(ArrayList<Char>(), predicate)\n}\n\n/**\n *
Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that
takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the
element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <T, C :
MutableCollection<in T>> Array<out T>.filterIndexedTo(destination: C, predicate: (index: Int, T) -> Boolean): C
{\n  forEachIndexed { index, element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n  return
destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n *
@param [predicate] function that takes the index of an element and the element itself\n * and returns the result of
predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\n
public inline fun <C : MutableCollection<in Byte>> ByteArray.filterIndexedTo(destination: C, predicate:
(index: Int, Byte) -> Boolean): C {\n  forEachIndexed { index, element ->\n    if (predicate(index, element))
destination.add(element)\n  }\n  return destination\n}\n\n/**\n * Appends all elements matching the given
[predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the
element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Short>>
ShortArray.filterIndexedTo(destination: C, predicate: (index: Int, Short) -> Boolean): C {\n  forEachIndexed {
index, element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n  return
destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n *
@param [predicate] function that takes the index of an element and the element itself\n * and returns the result of
predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\n
public inline fun <C : MutableCollection<in Int>> IntArray.filterIndexedTo(destination: C, predicate: (index:
Int, Int) -> Boolean): C {\n  forEachIndexed { index, element ->\n    if (predicate(index, element))
destination.add(element)\n  }\n  return destination\n}\n\n/**\n * Appends all elements matching the given
[predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the

```

```

element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Long>>
LongArray.filterIndexedTo(destination: C, predicate: (index: Int, Long) -> Boolean): C {\n  forEachIndexed {
index, element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n  return
destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n *
@param [predicate] function that takes the index of an element and the element itself\n * and returns the result of
predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n
*/\npublic inline fun <C : MutableCollection<in Float>> FloatArray.filterIndexedTo(destination: C, predicate:
(index: Int, Float) -> Boolean): C {\n  forEachIndexed { index, element ->\n    if (predicate(index, element))
destination.add(element)\n  }\n  return destination\n}\n\n/**\n * Appends all elements matching the given
[predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the
element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in
Double>> DoubleArray.filterIndexedTo(destination: C, predicate: (index: Int, Double) -> Boolean): C {\n
forEachIndexed { index, element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n  return
destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n *
@param [predicate] function that takes the index of an element and the element itself\n * and returns the result of
predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n
*/\npublic inline fun <C : MutableCollection<in Boolean>> BooleanArray.filterIndexedTo(destination: C, predicate:
(index: Int, Boolean) -> Boolean): C {\n  forEachIndexed { index, element ->\n    if (predicate(index, element))
destination.add(element)\n  }\n  return destination\n}\n\n/**\n * Appends all elements matching the given
[predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the
element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Char>>
CharArray.filterIndexedTo(destination: C, predicate: (index: Int, Char) -> Boolean): C {\n  forEachIndexed {
index, element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n  return
destination\n}\n\n/**\n * Returns a list containing all elements that are instances of specified type parameter R.\n *
\n * @sample samples.collections.Collections.Filtering.filterIsInstance\n */\npublic inline fun <reified R>
Array<*>.filterIsInstance(): List<@kotlin.internal.NoInfer R> {\n  return
filterIsInstanceTo(ArrayList<R>())\n}\n\n/**\n * Appends all elements that are instances of specified type
parameter R to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterIsInstanceTo\n */\npublic inline fun <reified R, C :
MutableCollection<in R>> Array<*>.filterIsInstanceTo(destination: C): C {\n  for (element in this) if (element is
R) destination.add(element)\n  return destination\n}\n\n/**\n * Returns a list containing all elements not matching
the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun <T>
Array<out T>.filterNot(predicate: (T) -> Boolean): List<T> {\n  return filterNotTo(ArrayList<T>(),
predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n */\npublic inline fun ByteArray.filterNot(predicate: (Byte) ->
Boolean): List<Byte> {\n  return filterNotTo(ArrayList<Byte>(), predicate)\n}\n\n/**\n * Returns a list containing
all elements not matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n
*/\npublic inline fun ShortArray.filterNot(predicate: (Short) -> Boolean): List<Short> {\n  return
filterNotTo(ArrayList<Short>(), predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the
given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun
IntArray.filterNot(predicate: (Int) -> Boolean): List<Int> {\n  return filterNotTo(ArrayList<Int>(),
predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n */\npublic inline fun LongArray.filterNot(predicate: (Long) ->
Boolean): List<Long> {\n  return filterNotTo(ArrayList<Long>(), predicate)\n}\n\n/**\n * Returns a list
containing all elements not matching the given [predicate].\n * \n * @sample

```

```

samples.collections.Collections.Filtering.filter\n *\npublic inline fun FloatArray.filterNot(predicate: (Float) ->
Boolean): List<Float> {\n  return filterNotTo(ArrayList<Float>(), predicate)\n}\n\n/**\n * Returns a list
containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n *\npublic inline fun DoubleArray.filterNot(predicate: (Double) ->
Boolean): List<Double> {\n  return filterNotTo(ArrayList<Double>(), predicate)\n}\n\n/**\n * Returns a list
containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n *\npublic inline fun BooleanArray.filterNot(predicate: (Boolean) -
> Boolean): List<Boolean> {\n  return filterNotTo(ArrayList<Boolean>(), predicate)\n}\n\n/**\n * Returns a list
containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n *\npublic inline fun CharArray.filterNot(predicate: (Char) ->
Boolean): List<Char> {\n  return filterNotTo(ArrayList<Char>(), predicate)\n}\n\n/**\n * Returns a list containing
all elements that are not `null`.\n * \n * @sample
samples.collections.Collections.Filtering.filterNotNull\n *\npublic
fun <T : Any> Array<out T?>.filterNotNull(): List<T> {\n  return filterNotNullTo(ArrayList<T>())\n}\n\n/**\n *
Appends all elements that are not `null` to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterNotNullTo\n *\npublic fun <C : MutableCollection<in T>, T : Any>
Array<out T?>.filterNotNullTo(destination: C): C {\n  for (element in this) if (element != null)
destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements not matching the given
[predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n *\npublic inline fun <T, C : MutableCollection<in T>> Array<out T>.filterNotTo(destination: C, predicate: (T) ->
Boolean): C {\n  for (element in this) if (!predicate(element)) destination.add(element)\n  return
destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n *
@sample
samples.collections.Collections.Filtering.filterTo\n *\npublic inline fun <C : MutableCollection<in
Byte>> ByteArray.filterNotTo(destination: C, predicate: (Byte) -> Boolean): C {\n  for (element in this) if
(!predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements not
matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n *\npublic inline fun <C : MutableCollection<in Short>>
ShortArray.filterNotTo(destination: C, predicate: (Short) -> Boolean): C {\n  for (element in this) if
(!predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements not
matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n *\npublic inline fun <C : MutableCollection<in Int>>
IntArray.filterNotTo(destination: C, predicate: (Int) -> Boolean): C {\n  for (element in this) if
(!predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements not
matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n *\npublic inline fun <C : MutableCollection<in Long>>
LongArray.filterNotTo(destination: C, predicate: (Long) -> Boolean): C {\n  for (element in this) if
(!predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements not
matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n *\npublic inline fun <C : MutableCollection<in Float>>
FloatArray.filterNotTo(destination: C, predicate: (Float) -> Boolean): C {\n  for (element in this) if
(!predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements not
matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n *\npublic inline fun <C : MutableCollection<in Double>>
DoubleArray.filterNotTo(destination: C, predicate: (Double) -> Boolean): C {\n  for (element in this) if
(!predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements not
matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n *\npublic inline fun <C : MutableCollection<in Boolean>>
BooleanArray.filterNotTo(destination: C, predicate: (Boolean) -> Boolean): C {\n  for (element in this) if
(!predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements not

```

```

matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n * \n public inline fun <C : MutableCollection<in Char>>
CharArray.filterNotTo(destination: C, predicate: (Char) -> Boolean): C {\n for (element in this) if
(!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements matching
the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n
*\n public inline fun <T, C : MutableCollection<in T>> Array<out T>.filterTo(destination: C, predicate: (T) ->
Boolean): C {\n for (element in this) if (predicate(element)) destination.add(element)\n return
destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * \n *
@sample samples.collections.Collections.Filtering.filterTo\n * \n public inline fun <C : MutableCollection<in
Byte>> ByteArray.filterTo(destination: C, predicate: (Byte) -> Boolean): C {\n for (element in this) if
(predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements matching
the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n
*\n public inline fun <C : MutableCollection<in Short>> ShortArray.filterTo(destination: C, predicate: (Short) ->
Boolean): C {\n for (element in this) if (predicate(element)) destination.add(element)\n return
destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * \n *
@sample samples.collections.Collections.Filtering.filterTo\n * \n public inline fun <C : MutableCollection<in Int>>
IntArray.filterTo(destination: C, predicate: (Int) -> Boolean): C {\n for (element in this) if (predicate(element))
destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to
the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n * \n public inline fun
<C : MutableCollection<in Long>> LongArray.filterTo(destination: C, predicate: (Long) -> Boolean): C {\n for
(element in this) if (predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all
elements matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n * \n public inline fun <C : MutableCollection<in Float>>
FloatArray.filterTo(destination: C, predicate: (Float) -> Boolean): C {\n for (element in this) if
(predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements matching
the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n
*\n public inline fun <C : MutableCollection<in Double>> DoubleArray.filterTo(destination: C, predicate: (Double)
-> Boolean): C {\n for (element in this) if (predicate(element)) destination.add(element)\n return
destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * \n *
@sample samples.collections.Collections.Filtering.filterTo\n * \n public inline fun <C : MutableCollection<in
Boolean>> BooleanArray.filterTo(destination: C, predicate: (Boolean) -> Boolean): C {\n for (element in this) if
(predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements matching
the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n
*\n public inline fun <C : MutableCollection<in Char>> CharArray.filterTo(destination: C, predicate: (Char) ->
Boolean): C {\n for (element in this) if (predicate(element)) destination.add(element)\n return
destination\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n * \n public
fun <T> Array<out T>.slice(indices: IntRange): List<T> {\n if (indices.isEmpty()) return listOf()\n return
copyOfRange(indices.start, indices.endInclusive + 1).asList()\n}\n\n/**\n * Returns a list containing elements at
indices in the specified [indices] range.\n * \n public fun ByteArray.slice(indices: IntRange): List<Byte> {\n if
(indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive +
1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n * \n public
fun ShortArray.slice(indices: IntRange): List<Short> {\n if (indices.isEmpty()) return listOf()\n return
copyOfRange(indices.start, indices.endInclusive + 1).asList()\n}\n\n/**\n * Returns a list containing elements at
indices in the specified [indices] range.\n * \n public fun IntArray.slice(indices: IntRange): List<Int> {\n if
(indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive +
1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n * \n public
fun LongArray.slice(indices: IntRange): List<Long> {\n if (indices.isEmpty()) return listOf()\n return
copyOfRange(indices.start, indices.endInclusive + 1).asList()\n}\n\n/**\n * Returns a list containing elements at

```



```

indices in the specified [indices] range.\n */\npublic fun FloatArray.slice(indices: IntRange): List<Float> {\n    if
(indices.isEmpty()) return listOf()\n    return copyOfRange(indices.start, indices.endInclusive +
1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n */\npublic
fun DoubleArray.slice(indices: IntRange): List<Double> {\n    if (indices.isEmpty()) return listOf()\n    return
copyOfRange(indices.start, indices.endInclusive + 1).asList()\n}\n\n/**\n * Returns a list containing elements at
indices in the specified [indices] range.\n */\npublic fun BooleanArray.slice(indices: IntRange): List<Boolean> {\n
if (indices.isEmpty()) return listOf()\n    return copyOfRange(indices.start, indices.endInclusive +
1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n */\npublic
fun CharArray.slice(indices: IntRange): List<Char> {\n    if (indices.isEmpty()) return listOf()\n    return
copyOfRange(indices.start, indices.endInclusive + 1).asList()\n}\n\n/**\n * Returns a list containing elements at
specified [indices].\n */\npublic fun <T> Array<out T>.slice(indices: Iterable<Int>): List<T> {\n    val size =
indices.collectionSizeOrDefault(10)\n    if (size == 0) return emptyList()\n    val list = ArrayList<T>(size)\n    for
(index in indices) {\n        list.add(get(index))\n    }\n    return list\n}\n\n/**\n * Returns a list containing elements at
specified [indices].\n */\npublic fun ByteArray.slice(indices: Iterable<Int>): List<Byte> {\n    val size =
indices.collectionSizeOrDefault(10)\n    if (size == 0) return emptyList()\n    val list = ArrayList<Byte>(size)\n    for
(index in indices) {\n        list.add(get(index))\n    }\n    return list\n}\n\n/**\n * Returns a list containing elements at
specified [indices].\n */\npublic fun ShortArray.slice(indices: Iterable<Int>): List<Short> {\n    val size =
indices.collectionSizeOrDefault(10)\n    if (size == 0) return emptyList()\n    val list = ArrayList<Short>(size)\n
for (index in indices) {\n        list.add(get(index))\n    }\n    return list\n}\n\n/**\n * Returns a list containing
elements at specified [indices].\n */\npublic fun IntArray.slice(indices: Iterable<Int>): List<Int> {\n    val size =
indices.collectionSizeOrDefault(10)\n    if (size == 0) return emptyList()\n    val list = ArrayList<Int>(size)\n    for
(index in indices) {\n        list.add(get(index))\n    }\n    return list\n}\n\n/**\n * Returns a list containing elements at
specified [indices].\n */\npublic fun LongArray.slice(indices: Iterable<Int>): List<Long> {\n    val size =
indices.collectionSizeOrDefault(10)\n    if (size == 0) return emptyList()\n    val list = ArrayList<Long>(size)\n
for (index in indices) {\n        list.add(get(index))\n    }\n    return list\n}\n\n/**\n * Returns a list containing
elements at specified [indices].\n */\npublic fun FloatArray.slice(indices: Iterable<Int>): List<Float> {\n    val size =
indices.collectionSizeOrDefault(10)\n    if (size == 0) return emptyList()\n    val list = ArrayList<Float>(size)\n
for (index in indices) {\n        list.add(get(index))\n    }\n    return list\n}\n\n/**\n * Returns a list containing
elements at specified [indices].\n */\npublic fun DoubleArray.slice(indices: Iterable<Int>): List<Double> {\n    val
size = indices.collectionSizeOrDefault(10)\n    if (size == 0) return emptyList()\n    val list =
ArrayList<Double>(size)\n    for (index in indices) {\n        list.add(get(index))\n    }\n    return list\n}\n\n/**\n *
Returns a list containing elements at specified [indices].\n */\npublic fun BooleanArray.slice(indices: Iterable<Int>):
List<Boolean> {\n    val size = indices.collectionSizeOrDefault(10)\n    if (size == 0) return emptyList()\n    val list
= ArrayList<Boolean>(size)\n    for (index in indices) {\n        list.add(get(index))\n    }\n    return list\n}\n\n/**\n *
Returns a list containing elements at specified [indices].\n */\npublic fun CharArray.slice(indices: Iterable<Int>):
List<Char> {\n    val size = indices.collectionSizeOrDefault(10)\n    if (size == 0) return emptyList()\n    val list =
ArrayList<Char>(size)\n    for (index in indices) {\n        list.add(get(index))\n    }\n    return list\n}\n\n/**\n *
Returns an array containing elements of this array at specified [indices].\n */\npublic fun <T>
Array<T>.sliceArray(indices: Collection<Int>): Array<T> {\n    val result = arrayOfNulls(this, indices.size)\n    var
targetIndex = 0\n    for (sourceIndex in indices) {\n        result[targetIndex++] = this[sourceIndex]\n    }\n    return
result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun
ByteArray.sliceArray(indices: Collection<Int>): ByteArray {\n    val result = ByteArray(indices.size)\n    var
targetIndex = 0\n    for (sourceIndex in indices) {\n        result[targetIndex++] = this[sourceIndex]\n    }\n    return
result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun
ShortArray.sliceArray(indices: Collection<Int>): ShortArray {\n    val result = ShortArray(indices.size)\n    var
targetIndex = 0\n    for (sourceIndex in indices) {\n        result[targetIndex++] = this[sourceIndex]\n    }\n    return
result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun
IntArray.sliceArray(indices: Collection<Int>): IntArray {\n    val result = IntArray(indices.size)\n    var targetIndex

```

```

= 0\n for (sourceIndex in indices) {\n    result[targetIndex++] = this[sourceIndex]\n }\n return
result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n *\npublic fun
LongArray.sliceArray(indices: Collection<Int>): LongArray {\n    val result = LongArray(indices.size)\n    var
targetIndex = 0\n    for (sourceIndex in indices) {\n        result[targetIndex++] = this[sourceIndex]\n    }\n    return
result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n *\npublic fun
FloatArray.sliceArray(indices: Collection<Int>): FloatArray {\n    val result = FloatArray(indices.size)\n    var
targetIndex = 0\n    for (sourceIndex in indices) {\n        result[targetIndex++] = this[sourceIndex]\n    }\n    return
result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n *\npublic fun
DoubleArray.sliceArray(indices: Collection<Int>): DoubleArray {\n    val result = DoubleArray(indices.size)\n
var targetIndex = 0\n    for (sourceIndex in indices) {\n        result[targetIndex++] = this[sourceIndex]\n    }\n
return result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n *\npublic fun
BooleanArray.sliceArray(indices: Collection<Int>): BooleanArray {\n    val result = BooleanArray(indices.size)\n
var targetIndex = 0\n    for (sourceIndex in indices) {\n        result[targetIndex++] = this[sourceIndex]\n    }\n
return result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n *\npublic fun
CharArray.sliceArray(indices: Collection<Int>): CharArray {\n    val result = CharArray(indices.size)\n    var
targetIndex = 0\n    for (sourceIndex in indices) {\n        result[targetIndex++] = this[sourceIndex]\n    }\n    return
result\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n *\npublic fun
<T> Array<T>.sliceArray(indices: IntRange): Array<T> {\n    if (indices.isEmpty()) return copyOfRange(0, 0)\n
return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at
indices in the specified [indices] range.\n *\npublic fun ByteArray.sliceArray(indices: IntRange): ByteArray {\n    if
(indices.isEmpty()) return ByteArray(0)\n    return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n
* Returns an array containing elements at indices in the specified [indices] range.\n *\npublic fun
ShortArray.sliceArray(indices: IntRange): ShortArray {\n    if (indices.isEmpty()) return ShortArray(0)\n    return
copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at indices
in the specified [indices] range.\n *\npublic fun IntArray.sliceArray(indices: IntRange): IntArray {\n    if
(indices.isEmpty()) return IntArray(0)\n    return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n
* Returns an array containing elements at indices in the specified [indices] range.\n *\npublic fun
LongArray.sliceArray(indices: IntRange): LongArray {\n    if (indices.isEmpty()) return LongArray(0)\n    return
copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at indices
in the specified [indices] range.\n *\npublic fun FloatArray.sliceArray(indices: IntRange): FloatArray {\n    if
(indices.isEmpty()) return FloatArray(0)\n    return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n
* Returns an array containing elements at indices in the specified [indices] range.\n *\npublic fun
DoubleArray.sliceArray(indices: IntRange): DoubleArray {\n    if (indices.isEmpty()) return DoubleArray(0)\n    return
copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at
indices in the specified [indices] range.\n *\npublic fun BooleanArray.sliceArray(indices: IntRange): BooleanArray
{\n    if (indices.isEmpty()) return BooleanArray(0)\n    return copyOfRange(indices.start, indices.endInclusive +
1)\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n *\npublic fun
CharArray.sliceArray(indices: IntRange): CharArray {\n    if (indices.isEmpty()) return CharArray(0)\n    return
copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns a list containing first [n] elements.\n *\n
* @throws IllegalArgumentException if [n] is negative.\n *\n\n * @sample
samples.collections.Collections.Transformations.take\n *\npublic fun <T> Array<out T>.take(n: Int): List<T> {\n
require(n >= 0) {\n    "Requested element count $n is less than zero." }\n    if (n == 0) return emptyList()\n    if (n >=
size) return toList()\n    if (n == 1) return listOf(this[0])\n    var count = 0\n    val list = ArrayList<T>(n)\n    for
(item in this) {\n        list.add(item)\n        if (++count == n)\n            break\n    }\n    return list\n}\n\n/**\n
* Returns a list containing first [n] elements.\n *\n * @throws IllegalArgumentException if [n] is negative.\n *\n * @sample
samples.collections.Collections.Transformations.take\n *\npublic fun ByteArray.take(n: Int): List<Byte> {\n
require(n >= 0) {\n    "Requested element count $n is less than zero." }\n    if (n == 0) return emptyList()\n    if (n >=
size) return toList()\n    if (n == 1) return listOf(this[0])\n    var count = 0\n    val list = ArrayList<Byte>(n)\n    for

```

```

(item in this) {\n    list.add(item)\n    if (++count == n)\n        break\n    }\n    return list\n}\n\n/**\n * Returns
a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\n\npublic fun ShortArray.take(n: Int): List<Short> {\n
require(n >= 0) {\n    list.add(item)\n    if (++count == n)\n        break\n    }\n    return list\n}\n\n/**\n * Returns
a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\n\npublic fun IntArray.take(n: Int): List<Int> {\n
require(n >= 0) {\n    list.add(item)\n    if (++count == n)\n        break\n    }\n    return list\n}\n\n/**\n * Returns
a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\n\npublic fun LongArray.take(n: Int): List<Long> {\n
require(n >= 0) {\n    list.add(item)\n    if (++count == n)\n        break\n    }\n    return list\n}\n\n/**\n * Returns
a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\n\npublic fun FloatArray.take(n: Int): List<Float> {\n
require(n >= 0) {\n    list.add(item)\n    if (++count == n)\n        break\n    }\n    return list\n}\n\n/**\n * Returns
a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\n\npublic fun DoubleArray.take(n: Int): List<Double> {\n
require(n >= 0) {\n    list.add(item)\n    if (++count == n)\n        break\n    }\n    return list\n}\n\n/**\n * Returns
a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\n\npublic fun BooleanArray.take(n: Int):
List<Boolean> {\n    require(n >= 0) {\n    list.add(item)\n    if (++count == n)\n        break\n    }\n    return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if
[n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\n\npublic fun
CharArray.take(n: Int): List<Char> {\n    require(n >= 0) {\n    list.add(item)\n    if (++count == n)\n        break\n    }\n    return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\n\npublic fun <T> Array<out T>.takeLast(n: Int): List<T>
{\n    require(n >= 0) {\n    list.add(this[index])\n    return list\n}\n\n/**\n * Returns
a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\n\npublic fun ByteArray.takeLast(n: Int): List<Byte> {\n
require(n >= 0) {\n    list.add(this[index])\n    return list\n}\n\n/**\n * Returns a list containing last
[n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample

```

```

samples.collections.Collections.Transformations.take\n *^\npublic fun ShortArray.takeLast(n: Int): List<Short> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n val size =
size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list = ArrayList<Short>(n)\n
for (index in size - n until size)\n list.add(this[index])\n return list\n}\n\n/**\n * Returns a list containing last
[n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic fun IntArray.takeLast(n: Int): List<Int> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n val size =
size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list = ArrayList<Int>(n)\n
for (index in size - n until size)\n list.add(this[index])\n return list\n}\n\n/**\n * Returns a list containing last
[n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic fun LongArray.takeLast(n: Int): List<Long> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n val size =
size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list = ArrayList<Long>(n)\n
for (index in size - n until size)\n list.add(this[index])\n return list\n}\n\n/**\n * Returns a list containing last
[n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic fun FloatArray.takeLast(n: Int): List<Float> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n val size =
size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list = ArrayList<Float>(n)\n
for (index in size - n until size)\n list.add(this[index])\n return list\n}\n\n/**\n * Returns a list containing last
[n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic fun DoubleArray.takeLast(n: Int): List<Double>
{\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n val
size = size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list =
ArrayList<Double>(n)\n for (index in size - n until size)\n list.add(this[index])\n return list\n}\n\n/**\n *
Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n *
@sample samples.collections.Collections.Transformations.take\n *^\npublic fun BooleanArray.takeLast(n: Int):
List<Boolean> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return
emptyList()\n val size = size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list
= ArrayList<Boolean>(n)\n for (index in size - n until size)\n list.add(this[index])\n return list\n}\n\n/**\n *
Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n *
@sample samples.collections.Collections.Transformations.take\n *^\npublic fun CharArray.takeLast(n: Int):
List<Char> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return
emptyList()\n val size = size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list
= ArrayList<Char>(n)\n for (index in size - n until size)\n list.add(this[index])\n return list\n}\n\n/**\n *
Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic inline fun <T> Array<out
T>.takeLastWhile(predicate: (T) -> Boolean): List<T> {\n for (index in lastIndex downTo 0) {\n if
(!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n * Returns a
list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic inline fun ByteArray.takeLastWhile(predicate:
(Byte) -> Boolean): List<Byte> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n
return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing last elements
satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n *^\npublic
inline fun ShortArray.takeLastWhile(predicate: (Short) -> Boolean): List<Short> {\n for (index in lastIndex
downTo 0) {\n if (!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n return
toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic inline fun IntArray.takeLastWhile(predicate: (Int)
-> Boolean): List<Int> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return

```

```

drop(index + 1)\n    }\n    }\n    return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the
given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n *^\npublic inline fun
LongArray.takeLastWhile(predicate: (Long) -> Boolean): List<Long> {\n    for (index in lastIndex downTo 0) {\n
    if (!predicate(this[index])) {\n        return drop(index + 1)\n    }\n    }\n    return toList()\n}\n\n/**\n * Returns
a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic inline fun FloatArray.takeLastWhile(predicate:
(Float) -> Boolean): List<Float> {\n    for (index in lastIndex downTo 0) {\n    if (!predicate(this[index])) {\n
    return drop(index + 1)\n    }\n    }\n    return toList()\n}\n\n/**\n * Returns a list containing last elements
satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n *^\npublic
inline fun DoubleArray.takeLastWhile(predicate: (Double) -> Boolean): List<Double> {\n    for (index in lastIndex
downTo 0) {\n    if (!predicate(this[index])) {\n        return drop(index + 1)\n    }\n    }\n    return
toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic inline fun
BooleanArray.takeLastWhile(predicate: (Boolean) -> Boolean): List<Boolean> {\n    for (index in lastIndex
downTo 0) {\n    if (!predicate(this[index])) {\n        return drop(index + 1)\n    }\n    }\n    return
toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic inline fun CharArray.takeLastWhile(predicate:
(Char) -> Boolean): List<Char> {\n    for (index in lastIndex downTo 0) {\n    if (!predicate(this[index])) {\n
    return drop(index + 1)\n    }\n    }\n    return toList()\n}\n\n/**\n * Returns a list containing first elements
satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n *^\npublic
inline fun <T> Array<out T>.takeWhile(predicate: (T) -> Boolean): List<T> {\n    val list = ArrayList<T>()\n    for
(item in this) {\n    if (!predicate(item))\n        break\n    list.add(item)\n    }\n    return list\n}\n\n/**\n *
Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic inline fun ByteArray.takeWhile(predicate: (Byte)
-> Boolean): List<Byte> {\n    val list = ArrayList<Byte>()\n    for (item in this) {\n    if (!predicate(item))\n
break\n    list.add(item)\n    }\n    return list\n}\n\n/**\n * Returns a list containing first elements satisfying the
given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n *^\npublic inline fun
ShortArray.takeWhile(predicate: (Short) -> Boolean): List<Short> {\n    val list = ArrayList<Short>()\n    for (item
in this) {\n    if (!predicate(item))\n        break\n    list.add(item)\n    }\n    return list\n}\n\n/**\n * Returns a
list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic inline fun IntArray.takeWhile(predicate: (Int) ->
Boolean): List<Int> {\n    val list = ArrayList<Int>()\n    for (item in this) {\n    if (!predicate(item))\n
break\n    list.add(item)\n    }\n    return list\n}\n\n/**\n * Returns a list containing first elements satisfying the
given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n *^\npublic inline fun
LongArray.takeWhile(predicate: (Long) -> Boolean): List<Long> {\n    val list = ArrayList<Long>()\n    for (item
in this) {\n    if (!predicate(item))\n        break\n    list.add(item)\n    }\n    return list\n}\n\n/**\n * Returns a
list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic inline fun FloatArray.takeWhile(predicate:
(Float) -> Boolean): List<Float> {\n    val list = ArrayList<Float>()\n    for (item in this) {\n    if
(!predicate(item))\n        break\n    list.add(item)\n    }\n    return list\n}\n\n/**\n * Returns a list containing first
elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n
*^\npublic inline fun DoubleArray.takeWhile(predicate: (Double) -> Boolean): List<Double> {\n    val list =
ArrayList<Double>()\n    for (item in this) {\n    if (!predicate(item))\n        break\n    list.add(item)\n    }\n
return list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *^\npublic inline fun BooleanArray.takeWhile(predicate:
(Boolean) -> Boolean): List<Boolean> {\n    val list = ArrayList<Boolean>()\n    for (item in this) {\n    if
(!predicate(item))\n        break\n    list.add(item)\n    }\n    return list\n}\n\n/**\n * Returns a list containing first
elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n

```



```

[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin("1.4")\npublic fun
ShortArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex,
size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var reverseIndex =
toIndex - 1\n    for (index in fromIndex until midPoint) {\n        val tmp = this[index]\n        this[index] =
this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements of the
array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n *
@param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin("1.4")\npublic fun
IntArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex,
size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var reverseIndex =
toIndex - 1\n    for (index in fromIndex until midPoint) {\n        val tmp = this[index]\n        this[index] =
this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements of the
array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n *
@param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin("1.4")\npublic fun
LongArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex,
size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var reverseIndex =
toIndex - 1\n    for (index in fromIndex until midPoint) {\n        val tmp = this[index]\n        this[index] =
this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements of the
array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n *
@param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin("1.4")\npublic fun
FloatArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex,
size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var reverseIndex =
toIndex - 1\n    for (index in fromIndex until midPoint) {\n        val tmp = this[index]\n        this[index] =
this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements of the
array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n *
@param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin("1.4")\npublic fun
DoubleArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex,
size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var reverseIndex =
toIndex - 1\n    for (index in fromIndex until midPoint) {\n        val tmp = this[index]\n        this[index] =
this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements of the
array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n *
@param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin("1.4")\npublic fun
BooleanArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex,
toIndex, size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var
reverseIndex = toIndex - 1\n    for (index in fromIndex until midPoint) {\n        val tmp = this[index]\n
this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses
elements of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to
reverse.\n * \n * @param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws

```

```

IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin("1.4")\npublic
fun CharArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex,
toIndex, size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var
reverseIndex = toIndex - 1\n    for (index in fromIndex until midPoint) {\n        val tmp = this[index]\n
this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Returns a
list with elements in reversed order.\n */\npublic fun <T> Array<out T>.reversed(): List<T> {\n    if (isEmpty())
return emptyList()\n    val list = toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns a list with
elements in reversed order.\n */\npublic fun ByteArray.reversed(): List<Byte> {\n    if (isEmpty()) return
emptyList()\n    val list = toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements
in reversed order.\n */\npublic fun ShortArray.reversed(): List<Short> {\n    if (isEmpty()) return emptyList()\n    val
list = toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n
*/\npublic fun IntArray.reversed(): List<Int> {\n    if (isEmpty()) return emptyList()\n    val list = toMutableList()\n
list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n */\npublic fun
LongArray.reversed(): List<Long> {\n    if (isEmpty()) return emptyList()\n    val list = toMutableList()\n
list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n */\npublic fun
FloatArray.reversed(): List<Float> {\n    if (isEmpty()) return emptyList()\n    val list = toMutableList()\n
list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n */\npublic fun
DoubleArray.reversed(): List<Double> {\n    if (isEmpty()) return emptyList()\n    val list = toMutableList()\n
list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n */\npublic fun
BooleanArray.reversed(): List<Boolean> {\n    if (isEmpty()) return emptyList()\n    val list = toMutableList()\n
list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n */\npublic fun
CharArray.reversed(): List<Char> {\n    if (isEmpty()) return emptyList()\n    val list = toMutableList()\n
list.reverse()\n    return list\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\npublic
fun <T> Array<T>.reversedArray(): Array<T> {\n    if (isEmpty()) return this\n    val result = arrayOfNulls(this,
size)\n    val lastIndex = lastIndex\n    for (i in 0..lastIndex)\n        result[lastIndex - i] = this[i]\n    return
result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\npublic fun
ByteArray.reversedArray(): ByteArray {\n    if (isEmpty()) return this\n    val result = ByteArray(size)\n    val
lastIndex = lastIndex\n    for (i in 0..lastIndex)\n        result[lastIndex - i] = this[i]\n    return result\n}\n\n/**\n *
Returns an array with elements of this array in reversed order.\n */\npublic fun ShortArray.reversedArray():
ShortArray {\n    if (isEmpty()) return this\n    val result = ShortArray(size)\n    val lastIndex = lastIndex\n    for (i in
0..lastIndex)\n        result[lastIndex - i] = this[i]\n    return result\n}\n\n/**\n * Returns an array with elements of this
array in reversed order.\n */\npublic fun IntArray.reversedArray(): IntArray {\n    if (isEmpty()) return this\n    val
result = IntArray(size)\n    val lastIndex = lastIndex\n    for (i in 0..lastIndex)\n        result[lastIndex - i] = this[i]\n
return result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\npublic fun
LongArray.reversedArray(): LongArray {\n    if (isEmpty()) return this\n    val result = LongArray(size)\n    val
lastIndex = lastIndex\n    for (i in 0..lastIndex)\n        result[lastIndex - i] = this[i]\n    return result\n}\n\n/**\n *
Returns an array with elements of this array in reversed order.\n */\npublic fun FloatArray.reversedArray():
FloatArray {\n    if (isEmpty()) return this\n    val result = FloatArray(size)\n    val lastIndex = lastIndex\n    for (i in
0..lastIndex)\n        result[lastIndex - i] = this[i]\n    return result\n}\n\n/**\n * Returns an array with elements of this
array in reversed order.\n */\npublic fun DoubleArray.reversedArray(): DoubleArray {\n    if (isEmpty()) return
this\n    val result = DoubleArray(size)\n    val lastIndex = lastIndex\n    for (i in 0..lastIndex)\n        result[lastIndex -
i] = this[i]\n    return result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\npublic
fun BooleanArray.reversedArray(): BooleanArray {\n    if (isEmpty()) return this\n    val result =
BooleanArray(size)\n    val lastIndex = lastIndex\n    for (i in 0..lastIndex)\n        result[lastIndex - i] = this[i]\n
return result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\npublic fun
CharArray.reversedArray(): CharArray {\n    if (isEmpty()) return this\n    val result = CharArray(size)\n    val
lastIndex = lastIndex\n    for (i in 0..lastIndex)\n        result[lastIndex - i] = this[i]\n    return result\n}\n\n/**\n *
Returns an array with elements of this array in reversed order.\n */\npublic fun <T> Array<T>.reversedArray(): Array<T> {\n
    if (isEmpty()) return this\n    val result = arrayOfNulls(this, size)\n    val lastIndex = lastIndex\n    for (i in
0..lastIndex)\n        result[lastIndex - i] = this[i]\n    return result\n}\n\n/**\n * Returns an array with elements of
this array in reversed order.\n */\npublic fun <T> Array<T>.reversedArray(): Array<T> {\n    if (isEmpty()) return
this\n    val result = arrayOfNulls(this, size)\n    val lastIndex = lastIndex\n    for (i in 0..lastIndex)\n        result[
lastIndex - i] = this[i]\n    return result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n
*/\npublic fun <T> Array<T>.reversedArray(): Array<T> {\n    if (isEmpty()) return this\n    val result = arrayOfNulls(this,
size)\n    val lastIndex = lastIndex\n    for (i in 0..lastIndex)\n        result[lastIndex - i] = this[i]\n    return
result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\npublic fun <T> Array<T>.reversedArray():
Array<T> {\n    if (isEmpty()) return this\n    val result = arrayOfNulls(this, size)\n    val lastIndex = lastIndex\n
    for (i in 0..lastIndex)\n        result[lastIndex - i] = this[i]\n    return result\n}\n\n/**\n * Returns an array with
elements of this array in reversed order.\n */\npublic fun <T> Array<T>.reversedArray(): Array<T> {\n    if (isEmpty())
return this\n    val result = arrayOfNulls(this, size)\n    val lastIndex = lastIndex\n    for (i in 0..lastIndex)\n       
result[lastIndex - i] = this[i]\n    return result\n}\n\n/**\n * Returns an array with elements of this array in reversed
order.\n */\npublic fun <T> Array<T>.reversedArray(): Array<T> {\n    if (isEmpty()) return this\n    val result =
arrayOfNulls(this, size)\n    val lastIndex = lastIndex\n    for (i in 0..lastIndex)\n        result[lastIndex - i] =
this[i]\n    return result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\npublic fun
<T> Array<T>.reversedArray(): Array<T> {\n    if (isEmpty()) return this\n    val result = arrayOfNulls(this, size)\n
    val lastIndex = lastIndex\n    for (i in 0..lastIndex)\n        result[lastIndex - i] = this[i]\n    return result\n}
}

```


Randomly shuffles elements in this array in-place.

```

*^@SinceKotlin("1.4")\npublic fun <T>
Array<T>.shuffle(): Unit {\n  shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n *^@SinceKotlin("1.4")\npublic fun ByteArray.shuffle(): Unit {\n  shuffle(Random)\n}\n\n/**\n * Randomly
shuffles elements in this array in-place.\n *^@SinceKotlin("1.4")\npublic fun ShortArray.shuffle(): Unit {\n
shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n
*^@SinceKotlin("1.4")\npublic fun IntArray.shuffle(): Unit {\n  shuffle(Random)\n}\n\n/**\n * Randomly
shuffles elements in this array in-place.\n *^@SinceKotlin("1.4")\npublic fun LongArray.shuffle(): Unit {\n
shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n
*^@SinceKotlin("1.4")\npublic fun FloatArray.shuffle(): Unit {\n  shuffle(Random)\n}\n\n/**\n * Randomly
shuffles elements in this array in-place.\n *^@SinceKotlin("1.4")\npublic fun DoubleArray.shuffle(): Unit {\n
shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n
*^@SinceKotlin("1.4")\npublic fun BooleanArray.shuffle(): Unit {\n  shuffle(Random)\n}\n\n/**\n * Randomly
shuffles elements in this array in-place.\n *^@SinceKotlin("1.4")\npublic fun CharArray.shuffle(): Unit {\n
shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random]
instance as the source of randomness.\n * \n * See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm\n
*^@SinceKotlin("1.4")\npublic fun <T> Array<T>.shuffle(random: Random): Unit {\n  for (i in lastIndex
downTo 1) {\n    val j = random.nextInt(i + 1)\n    val copy = this[i]\n    this[i] = this[j]\n    this[j] = copy\n
}\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the
source of randomness.\n * \n * See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm\n
*^@SinceKotlin("1.4")\npublic fun ByteArray.shuffle(random: Random): Unit {\n  for (i in lastIndex
downTo 1) {\n    val j = random.nextInt(i + 1)\n    val copy = this[i]\n    this[i] = this[j]\n    this[j] = copy\n
}\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the
source of randomness.\n * \n * See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm\n
*^@SinceKotlin("1.4")\npublic fun ShortArray.shuffle(random: Random): Unit {\n  for (i in lastIndex
downTo 1) {\n    val j = random.nextInt(i + 1)\n    val copy = this[i]\n    this[i] = this[j]\n    this[j] = copy\n
}\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the
source of randomness.\n * \n * See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm\n
*^@SinceKotlin("1.4")\npublic fun IntArray.shuffle(random: Random): Unit {\n  for (i in lastIndex
downTo 1) {\n    val j = random.nextInt(i + 1)\n    val copy = this[i]\n    this[i] = this[j]\n    this[j] = copy\n
}\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the
source of randomness.\n * \n * See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm\n
*^@SinceKotlin("1.4")\npublic fun LongArray.shuffle(random: Random): Unit {\n  for (i in lastIndex
downTo 1) {\n    val j = random.nextInt(i + 1)\n    val copy = this[i]\n    this[i] = this[j]\n    this[j] = copy\n
}\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the
source of randomness.\n * \n * See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm\n
*^@SinceKotlin("1.4")\npublic fun FloatArray.shuffle(random: Random): Unit {\n  for (i in lastIndex
downTo 1) {\n    val j = random.nextInt(i + 1)\n    val copy = this[i]\n    this[i] = this[j]\n    this[j] = copy\n
}\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the
source of randomness.\n * \n * See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm\n
*^@SinceKotlin("1.4")\npublic fun DoubleArray.shuffle(random: Random): Unit {\n  for (i in lastIndex
downTo 1) {\n    val j = random.nextInt(i + 1)\n    val copy = this[i]\n    this[i] = this[j]\n    this[j] = copy\n
}\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the
source of randomness.\n * \n * See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm\n

```

```

}
}

Randomly shuffles elements in this array in-place using the specified [random] instance as the
source of randomness. See:
https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates\_shuffle#The\_modern\_algorithm

@SinceKotlin("1.4")
public fun BooleanArray.shuffle(random: Random): Unit {
    for (i in lastIndex
        downTo 1) {
        val j = random.nextInt(i + 1)
        val copy = this[i]
        this[i] = this[j]
        this[j] = copy
    }
}

Randomly shuffles elements in this array in-place using the specified [random] instance as the
source of randomness. See:
https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates\_shuffle#The\_modern\_algorithm

@SinceKotlin("1.4")
public fun CharArray.shuffle(random: Random): Unit {
    for (i in lastIndex downTo
        1) {
        val j = random.nextInt(i + 1)
        val copy = this[i]
        this[i] = this[j]
        this[j] = copy
    }
}

Sorts elements in the array in-place according to natural sort order of the value returned by specified
[selector] function. The sort is _stable_. It means that equal elements preserve their order relative to each
other after sorting.

public inline fun <T, R : Comparable<R>> Array<out T>.sortBy(crossinline selector: (T) -
    > R?): Unit {
    if (size > 1) sortWith(compareBy(selector))
}

Sorts elements in the array in-place
descending according to natural sort order of the value returned by specified [selector] function. The sort is
_stable_. It means that equal elements preserve their order relative to each other after sorting.

public inline fun
<T, R : Comparable<R>> Array<out T>.sortByDescending(crossinline selector: (T) -> R?): Unit {
    if (size > 1)
        sortWith(compareByDescending(selector))
}

Sorts elements in the array in-place descending according
to their natural sort order. The sort is _stable_. It means that equal elements preserve their order relative to
each other after sorting.

public fun <T : Comparable<T>> Array<out T>.sortDescending(): Unit {
    sortWith(reverseOrder())
}

Sorts elements in the array in-place descending according to their natural
sort order.

public fun ByteArray.sortDescending(): Unit {
    if (size > 1) {
        sort()
        reverse()
    }
}

Sorts elements in the array in-place descending according to their natural sort order.

public fun ShortArray.sortDescending(): Unit {
    if (size > 1) {
        sort()
        reverse()
    }
}

Sorts
elements in the array in-place descending according to their natural sort order.

public fun
IntArray.sortDescending(): Unit {
    if (size > 1) {
        sort()
        reverse()
    }
}

Sorts elements
in the array in-place descending according to their natural sort order.

public fun LongArray.sortDescending():
    Unit {
    if (size > 1) {
        sort()
        reverse()
    }
}

Sorts elements in the array in-place
descending according to their natural sort order.

public fun FloatArray.sortDescending(): Unit {
    if (size >
        1) {
        sort()
        reverse()
    }
}

Sorts elements in the array in-place descending according to
their natural sort order.

public fun DoubleArray.sortDescending(): Unit {
    if (size > 1) {
        sort()
        reverse()
    }
}

Sorts elements in the array in-place descending according to their natural sort order.

public fun CharArray.sortDescending(): Unit {
    if (size > 1) {
        sort()
        reverse()
    }
}

Returns a list of all elements sorted according to their natural sort order. The sort is _stable_. It means that
equal elements preserve their order relative to each other after sorting.

public fun <T : Comparable<T>>
    Array<out T>.sorted(): List<T> {
    return sortedArray().asList()
}

Returns a list of all elements sorted
according to their natural sort order.

public fun ByteArray.sorted(): List<Byte> {
    return
        toTypedArray().apply { sort() }.asList()
}

Returns a list of all elements sorted according to their natural
sort order.

public fun ShortArray.sorted(): List<Short> {
    return toTypedArray().apply { sort()
        }.asList()
}

Returns a list of all elements sorted according to their natural sort order.

public fun
    IntArray.sorted(): List<Int> {
    return toTypedArray().apply { sort() }.asList()
}

Returns a list of all
elements sorted according to their natural sort order.

public fun LongArray.sorted(): List<Long> {
    return
        toTypedArray().apply { sort() }.asList()
}

Returns a list of all elements sorted according to their natural
sort order.

public fun FloatArray.sorted(): List<Float> {
    return toTypedArray().apply { sort()
        }.asList()
}

Returns a list of all elements sorted according to their natural sort order.

public fun
    DoubleArray.sorted(): List<Double> {
    return toTypedArray().apply { sort() }.asList()
}

Returns a
list of all elements sorted according to their natural sort order.

public fun CharArray.sorted(): List<Char> {
    return
        toTypedArray().apply { sort() }.asList()
}

Returns an array with all elements of this array sorted

```

according to their natural sort order.\n * \n * The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T : Comparable<T>> Array<T>.sortedArray(): Array<T> {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\npublic fun ByteArray.sortedArray(): ByteArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\npublic fun ShortArray.sortedArray(): ShortArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\npublic fun IntArray.sortedArray(): IntArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\npublic fun LongArray.sortedArray(): LongArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\npublic fun FloatArray.sortedArray(): FloatArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\npublic fun DoubleArray.sortedArray(): DoubleArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\npublic fun CharArray.sortedArray(): CharArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n * \n * The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T : Comparable<T>> Array<T>.sortedArrayDescending(): Array<T> {\n if (isEmpty()) return this\n return this.copyOf().apply { sortWith(reverseOrder()) }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n */\npublic fun ByteArray.sortedArrayDescending(): ByteArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n */\npublic fun ShortArray.sortedArrayDescending(): ShortArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n */\npublic fun IntArray.sortedArrayDescending(): IntArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n */\npublic fun LongArray.sortedArrayDescending(): LongArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n */\npublic fun FloatArray.sortedArrayDescending(): FloatArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n */\npublic fun DoubleArray.sortedArrayDescending(): DoubleArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n */\npublic fun CharArray.sortedArrayDescending(): CharArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according the specified [comparator].\n * \n * The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T> Array<out T>.sortedArrayWith(comparator: Comparator<in T>): Array<out T> {\n if (isEmpty()) return this\n return this.copyOf().apply { sortWith(comparator) }\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n * \n * The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n * \n * @sample samples.collections.Collections.Sorting.sortedBy\n */\npublic inline fun <T, R : Comparable<R>> Array<out T>.sortedBy(crossinline selector: (T) -> R?): List<T> {\n return sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n * \n * @sample samples.collections.Collections.Sorting.sortedBy\n */\npublic inline fun <R : Comparable<R>>

ByteArray.sortedBy(crossinline selector: (Byte) -> R?): List<Byte> {\n return
 sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order
 of the value returned by specified [selector] function.\n * \n * @sample
 samples.collections.Collections.Sorting.sortedBy\n * \n\npublic inline fun <R : Comparable<R>>

ShortArray.sortedBy(crossinline selector: (Short) -> R?): List<Short> {\n return
 sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order
 of the value returned by specified [selector] function.\n * \n * @sample
 samples.collections.Collections.Sorting.sortedBy\n * \n\npublic inline fun <R : Comparable<R>>

IntArray.sortedBy(crossinline selector: (Int) -> R?): List<Int> {\n return
 sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order
 of the value returned by specified [selector] function.\n * \n * @sample
 samples.collections.Collections.Sorting.sortedBy\n * \n\npublic inline fun <R : Comparable<R>>

LongArray.sortedBy(crossinline selector: (Long) -> R?): List<Long> {\n return
 sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order
 of the value returned by specified [selector] function.\n * \n * @sample
 samples.collections.Collections.Sorting.sortedBy\n * \n\npublic inline fun <R : Comparable<R>>

FloatArray.sortedBy(crossinline selector: (Float) -> R?): List<Float> {\n return
 sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order
 of the value returned by specified [selector] function.\n * \n * @sample
 samples.collections.Collections.Sorting.sortedBy\n * \n\npublic inline fun <R : Comparable<R>>

DoubleArray.sortedBy(crossinline selector: (Double) -> R?): List<Double> {\n return
 sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order
 of the value returned by specified [selector] function.\n * \n * @sample
 samples.collections.Collections.Sorting.sortedBy\n * \n\npublic inline fun <R : Comparable<R>>

BooleanArray.sortedBy(crossinline selector: (Boolean) -> R?): List<Boolean> {\n return
 sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order
 of the value returned by specified [selector] function.\n * \n * @sample
 samples.collections.Collections.Sorting.sortedBy\n * \n\npublic inline fun <R : Comparable<R>>

CharArray.sortedBy(crossinline selector: (Char) -> R?): List<Char> {\n return
 sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural
 sort order of the value returned by specified [selector] function.\n * \n * The sort is `_stable_`. It means that equal
 elements preserve their order relative to each other after sorting.\n * \n\npublic inline fun <T, R : Comparable<R>>

Array<out T>.sortedByDescending(crossinline selector: (T) -> R?): List<T> {\n return
 sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending
 according to natural sort order of the value returned by specified [selector] function.\n * \n\npublic inline fun <R :
 Comparable<R>> ByteArray.sortedByDescending(crossinline selector: (Byte) -> R?): List<Byte> {\n return
 sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending
 according to natural sort order of the value returned by specified [selector] function.\n * \n\npublic inline fun <R :
 Comparable<R>> ShortArray.sortedByDescending(crossinline selector: (Short) -> R?): List<Short> {\n return
 sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending
 according to natural sort order of the value returned by specified [selector] function.\n * \n\npublic inline fun <R :
 Comparable<R>> IntArray.sortedByDescending(crossinline selector: (Int) -> R?): List<Int> {\n return
 sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending
 according to natural sort order of the value returned by specified [selector] function.\n * \n\npublic inline fun <R :
 Comparable<R>> LongArray.sortedByDescending(crossinline selector: (Long) -> R?): List<Long> {\n return
 sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending
 according to natural sort order of the value returned by specified [selector] function.\n * \n\npublic inline fun <R :
 Comparable<R>> FloatArray.sortedByDescending(crossinline selector: (Float) -> R?): List<Float> {\n return

sortedWith(compareByDescending(selector))\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> DoubleArray.sortedByDescending(crossinline selector: (Double) -> R?): List<Double> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> BooleanArray.sortedByDescending(crossinline selector: (Boolean) -> R?): List<Boolean> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> CharArray.sortedByDescending(crossinline selector: (Char) -> R?): List<Char> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n * \n * The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T : Comparable<T>> Array<out T>.sortedDescending(): List<T> {\n return sortedWith(reverseOrder())\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun ByteArray.sortedDescending(): List<Byte> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun ShortArray.sortedDescending(): List<Short> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun IntArray.sortedDescending(): List<Int> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun LongArray.sortedDescending(): List<Long> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun FloatArray.sortedDescending(): List<Float> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun DoubleArray.sortedDescending(): List<Double> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun CharArray.sortedDescending(): List<Char> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n * \n * The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T> Array<out T>.sortedWith(comparator: Comparator<in T>): List<T> {\n return sortedArrayWith(comparator).asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun ByteArray.sortedWith(comparator: Comparator<in Byte>): List<Byte> {\n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun ShortArray.sortedWith(comparator: Comparator<in Short>): List<Short> {\n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun IntArray.sortedWith(comparator: Comparator<in Int>): List<Int> {\n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun LongArray.sortedWith(comparator: Comparator<in Long>): List<Long> {\n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun FloatArray.sortedWith(comparator: Comparator<in Float>): List<Float> {\n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun DoubleArray.sortedWith(comparator: Comparator<in Double>): List<Double> {\n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun BooleanArray.sortedWith(comparator: Comparator<in Boolean>): List<Boolean> {\n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun CharArray.sortedWith(comparator: Comparator<in Char>): List<Char> {\n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\npublic expect fun

`<T> Array<out T>.asList(): List<T>` Returns a [List] that wraps the original array.

`fun ByteArray.asList(): List<Byte>` Returns a [List] that wraps the original array.

`fun ShortArray.asList(): List<Short>` Returns a [List] that wraps the original array.

`fun IntArray.asList(): List<Int>` Returns a [List] that wraps the original array.

`fun LongArray.asList(): List<Long>` Returns a [List] that wraps the original array.

`fun FloatArray.asList(): List<Float>` Returns a [List] that wraps the original array.

`fun DoubleArray.asList(): List<Double>` Returns a [List] that wraps the original array.

`fun BooleanArray.asList(): List<Boolean>` Returns a [List] that wraps the original array.

`fun CharArray.asList(): List<Char>` Returns `true` if the two specified arrays are *deeply* equal to one another, i.e. contain the same number of the same elements in the same order. If two corresponding elements are nested arrays, they are also compared deeply. If any of arrays contains itself on any nesting level the behavior is undefined. The elements of other types are compared for equality with the `[equals][Any.equals]` function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

`@SinceKotlin("1.1")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic expect infix fun <T> Array<out T>.contentDeepEquals(other: Array<out T>): Boolean` Returns `true` if the two specified arrays are *deeply* equal to one another, i.e. contain the same number of the same elements in the same order. The specified arrays are also considered deeply equal if both are `null`. If two corresponding elements are nested arrays, they are also compared deeply. If any of arrays contains itself on any nesting level the behavior is undefined. The elements of other types are compared for equality with the `[equals][Any.equals]` function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

`@SinceKotlin("1.4")\npublic expect infix fun <T> Array<out T>?.contentDeepEquals(other: Array<out T>?): Boolean` Returns a hash code based on the contents of this array as if it is [List]. Nested arrays are treated as lists too. If any of arrays contains itself on any nesting level the behavior is undefined.

`@SinceKotlin("1.1")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic expect fun <T> Array<out T>.contentDeepHashCode(): Int` Returns a hash code based on the contents of this array as if it is [List]. Nested arrays are treated as lists too. If any of arrays contains itself on any nesting level the behavior is undefined.

`@SinceKotlin("1.4")\npublic expect fun <T> Array<out T>?.contentDeepHashCode(): Int` Returns a string representation of the contents of this array as if it is a [List]. Nested arrays are treated as lists too. If any of arrays contains itself on any nesting level that reference is rendered as `"[...]"` to prevent recursion.

`@sample samples.collections.Arrays.ContentOperations.contentDeepToString\n\n@SinceKotlin("1.1")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic expect fun <T> Array<out T>.contentDeepToString(): String` Returns a string representation of the contents of this array as if it is a [List]. Nested arrays are treated as lists too. If any of arrays contains itself on any nesting level that reference is rendered as `"[...]"` to prevent recursion.

`@sample samples.collections.Arrays.ContentOperations.contentDeepToString\n\n@SinceKotlin("1.4")\npublic expect fun <T> Array<out T>?.contentDeepToString(): String` Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the `[equals][Any.equals]` function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

`@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect infix fun <T> Array<out T>.contentEquals(other: Array<out T>): Boolean` Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the `[equals][Any.equals]` function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

`@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect infix fun ByteArray.contentEquals(other: ByteArray): Boolean` Returns `true` if the two specified arrays are


```

equal to `0.0`.
\n * \n @SinceKotlin("1.4")
\n public expect infix fun ShortArray?.contentEquals(other: ShortArray?):
Boolean
\n \n /**
\n * Returns `true` if the two specified arrays are *structurally* equal to one another,
\n * i.e. contain the same number of the same elements in the same order.
\n * \n * The elements are compared for equality with the [equals][Any.equals] function.
\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not
equal to `0.0`.
\n * \n @SinceKotlin("1.4")
\n public expect infix fun IntArray?.contentEquals(other: IntArray?):
Boolean
\n \n /**
\n * Returns `true` if the two specified arrays are *structurally* equal to one another,
\n * i.e. contain the same number of the same elements in the same order.
\n * \n * The elements are compared for equality with the [equals][Any.equals] function.
\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not
equal to `0.0`.
\n * \n @SinceKotlin("1.4")
\n public expect infix fun LongArray?.contentEquals(other: LongArray?):
Boolean
\n \n /**
\n * Returns `true` if the two specified arrays are *structurally* equal to one another,
\n * i.e. contain the same number of the same elements in the same order.
\n * \n * The elements are compared for equality with the [equals][Any.equals] function.
\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not
equal to `0.0`.
\n * \n @SinceKotlin("1.4")
\n public expect infix fun FloatArray?.contentEquals(other: FloatArray?):
Boolean
\n \n /**
\n * Returns `true` if the two specified arrays are *structurally* equal to one another,
\n * i.e. contain the same number of the same elements in the same order.
\n * \n * The elements are compared for equality with the [equals][Any.equals] function.
\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not
equal to `0.0`.
\n * \n @SinceKotlin("1.4")
\n public expect infix fun DoubleArray?.contentEquals(other:
DoubleArray?): Boolean
\n \n /**
\n * Returns `true` if the two specified arrays are *structurally* equal to one
another,
\n * i.e. contain the same number of the same elements in the same order.
\n * \n * The elements are compared for equality with the [equals][Any.equals] function.
\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not
equal to `0.0`.
\n * \n @SinceKotlin("1.4")
\n public expect infix fun BooleanArray?.contentEquals(other: BooleanArray?): Boolean
\n \n /**
\n * Returns `true` if the two specified arrays are *structurally* equal to one another,
\n * i.e. contain the same number of the same elements in the same order.
\n * \n * The elements are compared for equality with the [equals][Any.equals] function.
\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not
equal to `0.0`.
\n * \n @SinceKotlin("1.4")
\n public expect infix fun CharArray?.contentHashCode(): Int
\n \n /**
\n * Returns a hash code based on the contents of this array as if it is [List].
\n * \n @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")
\n @SinceKotlin("1.1")
\n @DeprecatedSinceKotlin(hiddenSince = "1.4")
\n public expect fun <T>
Array<out T>.contentHashCode(): Int
\n \n /**
\n * Returns a hash code based on the contents of this array as if it is [List].
\n * \n @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")
\n @SinceKotlin("1.1")
\n @DeprecatedSinceKotlin(hiddenSince = "1.4")
\n public expect fun ByteArray.contentHashCode(): Int
\n \n /**
\n * Returns a hash code based on the contents of this array as if it is [List].
\n * \n @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")
\n @SinceKotlin("1.1")
\n @DeprecatedSinceKotlin(hiddenSince = "1.4")
\n public expect fun ShortArray.contentHashCode(): Int
\n \n /**
\n * Returns a hash code based on the contents of this array as if it is [List].
\n * \n @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")
\n @SinceKotlin("1.1")
\n @DeprecatedSinceKotlin(hiddenSince = "1.4")
\n public expect fun IntArray.contentHashCode(): Int
\n \n /**
\n * Returns a hash code based on the contents of this array as if it is [List].
\n * \n @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")
\n @SinceKotlin("1.1")
\n @DeprecatedSinceKotlin(hiddenSince = "1.4")
\n public expect fun LongArray.contentHashCode(): Int
\n \n /**
\n * Returns a hash code based on the contents of this array as if it is [List].
\n * \n @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")
\n @SinceKotlin("1.1")
\n @DeprecatedSinceKotlin(hiddenSince = "1.4")
\n public expect fun FloatArray.contentHashCode(): Int
\n \n /**
\n * Returns a hash code based on the contents of this array as if it is [List].
\n * \n @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")
\n @SinceKotlin("1.1")
\n @DeprecatedSinceKotlin(hiddenSince = "1.4")
\n public expect fun DoubleArray.contentHashCode(): Int
\n \n /**
\n * Returns a hash code based on the contents of this array as if it is [List].
\n * \n @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation

```


warning.\n)\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun BooleanArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.\n")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun CharArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\npublic expect fun <T> Array<out T>?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\npublic expect fun ByteArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\npublic expect fun ShortArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\npublic expect fun IntArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\npublic expect fun LongArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\npublic expect fun FloatArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\npublic expect fun DoubleArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\npublic expect fun BooleanArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\npublic expect fun CharArray?.contentToString(): String\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin("1.3")\npublic expect fun <T> Array<out T>.copyInto(destination: Array<T>, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): Array<T>\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin("1.3")\npublic expect fun ByteArray.copyInto(destination: ByteArray,

destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ByteArray\n\n**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n\n*\n\n@SinceKotlin("1.3")\npublic expect fun ShortArray.copyInto(destination: ShortArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ShortArray\n\n**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n\n*\n\n@SinceKotlin("1.3")\npublic expect fun IntArray.copyInto(destination: IntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): IntArray\n\n**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n\n*\n\n@SinceKotlin("1.3")\npublic expect fun LongArray.copyInto(destination: LongArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): LongArray\n\n**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n\n*\n\n@SinceKotlin("1.3")\npublic expect fun FloatArray.copyInto(destination: FloatArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): FloatArray\n\n**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or

the original array, the extra elements in the copy array are filled with zero values.

`ByteArrayList.CopyOfOperations.resizedPrimitiveCopyOf` (public expect fun)

`ByteArray.copyOf(newSize: Int): ByteArray`

Returns new array which is a copy of the original array, resized to the given [newSize]. The copy is either truncated or padded at the end with zero values if necessary.

If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize]. If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.

`ShortArrayList.CopyOfOperations.resizedPrimitiveCopyOf` (public expect fun)

`ShortArray.copyOf(newSize: Int): ShortArray`

Returns new array which is a copy of the original array, resized to the given [newSize]. The copy is either truncated or padded at the end with zero values if necessary.

If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize]. If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.

`IntArray.CopyOfOperations.resizedPrimitiveCopyOf` (public expect fun)

`IntArray.copyOf(newSize: Int): IntArray`

Returns new array which is a copy of the original array, resized to the given [newSize]. The copy is either truncated or padded at the end with zero values if necessary.

If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize]. If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.

`LongArrayList.CopyOfOperations.resizedPrimitiveCopyOf` (public expect fun)

`LongArray.copyOf(newSize: Int): LongArray`

Returns new array which is a copy of the original array, resized to the given [newSize]. The copy is either truncated or padded at the end with zero values if necessary.

If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize]. If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.

`FloatArrayList.CopyOfOperations.resizedPrimitiveCopyOf` (public expect fun)

`FloatArray.copyOf(newSize: Int): FloatArray`

Returns new array which is a copy of the original array, resized to the given [newSize]. The copy is either truncated or padded at the end with zero values if necessary.

If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize]. If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.

`DoubleArrayList.CopyOfOperations.resizedPrimitiveCopyOf` (public expect fun)

`DoubleArray.copyOf(newSize: Int): DoubleArray`

Returns new array which is a copy of the original array, resized to the given [newSize]. The copy is either truncated or padded at the end with `false` values if necessary.

If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize]. If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with `false` values.

`BooleanArrayList.CopyOfOperations.resizedPrimitiveCopyOf` (public expect fun)

`BooleanArray.copyOf(newSize: Int): BooleanArray`

Returns new array which is a copy of the original array, resized to the given [newSize]. The copy is either truncated or padded at the end with null char (‘\u0000’) values if necessary.

If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize]. If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with null char (‘\u0000’) values.

`CharArray.CopyOfOperations.resizingCopyOf` (public expect fun)

`CharArray.copyOf(newSize: Int): CharArray`

Returns new array which is a copy of the original array, resized to the given [newSize]. The copy is either truncated or padded at the end with `null` values if necessary.

If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize]. If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with `null` values.

`Array.copyOf` (public expect fun)

`<T> Array<T>.copyOf(newSize: Int): Array<T>`

Returns a new array which is a copy of the specified range of the original array.

@param fromIndex the start of the range (inclusive) to copy.

@param toIndex the end of the range (exclusive) to copy.

@throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the

size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *
`@Suppress("NO_ACTUAL_FOR_EXPECT")\npublic expect fun <T> Array<T>.copyOfRange(fromIndex: Int, toIndex: Int): Array<T>`\n\n**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *
`\npublic expect fun ByteArray.copyOfRange(fromIndex: Int, toIndex: Int): ByteArray`\n\n**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *
`\npublic expect fun ShortArray.copyOfRange(fromIndex: Int, toIndex: Int): ShortArray`\n\n**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *
`\npublic expect fun IntArray.copyOfRange(fromIndex: Int, toIndex: Int): IntArray`\n\n**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *
`\npublic expect fun LongArray.copyOfRange(fromIndex: Int, toIndex: Int): LongArray`\n\n**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *
`\npublic expect fun FloatArray.copyOfRange(fromIndex: Int, toIndex: Int): FloatArray`\n\n**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *
`\npublic expect fun DoubleArray.copyOfRange(fromIndex: Int, toIndex: Int): DoubleArray`\n\n**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *
`\npublic expect fun BooleanArray.copyOfRange(fromIndex: Int, toIndex: Int): BooleanArray`\n\n**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *
`\npublic expect fun CharArray.copyOfRange(fromIndex: Int, toIndex: Int): CharArray`\n\n**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *
`@SinceKotlin("1.3")\npublic expect fun <T> Array<T>.fill(element: T, fromIndex: Int = 0, toIndex: Int = size): Unit`\n\n**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill,

0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

```

*\n@SinceKotlin("1.3")\npublic expect fun ByteArray.fill(element: Byte, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
```

```

*\n@SinceKotlin("1.3")\npublic expect fun ShortArray.fill(element: Short, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
```

```

*\n@SinceKotlin("1.3")\npublic expect fun IntArray.fill(element: Int, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
```

```

*\n@SinceKotlin("1.3")\npublic expect fun LongArray.fill(element: Long, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
```

```

*\n@SinceKotlin("1.3")\npublic expect fun FloatArray.fill(element: Float, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
```

```

*\n@SinceKotlin("1.3")\npublic expect fun DoubleArray.fill(element: Double, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
```

```

*\n@SinceKotlin("1.3")\npublic expect fun BooleanArray.fill(element: Boolean, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
```

```

*\n@SinceKotlin("1.3")\npublic expect fun CharArray.fill(element: Char, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Returns the range of valid indices for the array.\n *\npublic val <T> Array<out T>.indices: IntRange\n    get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid indices for the array.\n *\npublic val ByteArray.indices: IntRange\n    get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid indices for the array.\n *\npublic val ShortArray.indices: IntRange\n    get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid indices for the array.\n *\npublic val IntArray.indices: IntRange\n    get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid indices for the array.\n *\npublic val LongArray.indices: IntRange\n    get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid indices for the array.\n *\npublic val FloatArray.indices: IntRange\n    get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid indices for

```

the array.
`public val DoubleArray.indices: IntRange`
`get() = IntRange(0, lastIndex)`
Returns the range of valid indices for the array.
`public val BooleanArray.indices: IntRange`
`get() = IntRange(0, lastIndex)`
Returns the range of valid indices for the array.
`public val CharArray.indices: IntRange`
`get() = IntRange(0, lastIndex)`
Returns `true` if the array is empty.
`@kotlin.internal.InlineOnly`
`public inline fun <T> Array<out T>.isEmpty(): Boolean` {
return size == 0
}
Returns `true` if the array is empty.
`@kotlin.internal.InlineOnly`
`public inline fun ByteArray.isEmpty(): Boolean` {
return size == 0
}
Returns `true` if the array is empty.
`@kotlin.internal.InlineOnly`
`public inline fun ShortArray.isEmpty(): Boolean` {
return size == 0
}
Returns `true` if the array is empty.
`@kotlin.internal.InlineOnly`
`public inline fun IntArray.isEmpty(): Boolean` {
return size == 0
}
Returns `true` if the array is empty.
`@kotlin.internal.InlineOnly`
`public inline fun LongArray.isEmpty(): Boolean` {
return size == 0
}
Returns `true` if the array is empty.
`@kotlin.internal.InlineOnly`
`public inline fun FloatArray.isEmpty(): Boolean` {
return size == 0
}
Returns `true` if the array is empty.
`@kotlin.internal.InlineOnly`
`public inline fun DoubleArray.isEmpty(): Boolean` {
return size == 0
}
Returns `true` if the array is empty.
`@kotlin.internal.InlineOnly`
`public inline fun BooleanArray.isEmpty(): Boolean` {
return size == 0
}
Returns `true` if the array is empty.
`@kotlin.internal.InlineOnly`
`public inline fun CharArray.isEmpty(): Boolean` {
return size == 0
}
Returns `true` if the array is not empty.
`@kotlin.internal.InlineOnly`
`public inline fun <T> Array<out T>.isNotEmpty(): Boolean` {
return !isEmpty()
}
Returns `true` if the array is not empty.
`@kotlin.internal.InlineOnly`
`public inline fun ByteArray.isNotEmpty(): Boolean` {
return !isEmpty()
}
Returns `true` if the array is not empty.
`@kotlin.internal.InlineOnly`
`public inline fun ShortArray.isNotEmpty(): Boolean` {
return !isEmpty()
}
Returns `true` if the array is not empty.
`@kotlin.internal.InlineOnly`
`public inline fun IntArray.isNotEmpty(): Boolean` {
return !isEmpty()
}
Returns `true` if the array is not empty.
`@kotlin.internal.InlineOnly`
`public inline fun LongArray.isNotEmpty(): Boolean` {
return !isEmpty()
}
Returns `true` if the array is not empty.
`@kotlin.internal.InlineOnly`
`public inline fun FloatArray.isNotEmpty(): Boolean` {
return !isEmpty()
}
Returns `true` if the array is not empty.
`@kotlin.internal.InlineOnly`
`public inline fun DoubleArray.isNotEmpty(): Boolean` {
return !isEmpty()
}
Returns `true` if the array is not empty.
`@kotlin.internal.InlineOnly`
`public inline fun BooleanArray.isNotEmpty(): Boolean` {
return !isEmpty()
}
Returns `true` if the array is not empty.
`@kotlin.internal.InlineOnly`
`public inline fun CharArray.isNotEmpty(): Boolean` {
return !isEmpty()
}
Returns the last valid index for the array.
`public val <T> Array<out T>.lastIndex: Int`
`get() = size - 1`
Returns the last valid index for the array.
`public val ByteArray.lastIndex: Int`
`get() = size - 1`
Returns the last valid index for the array.
`public val ShortArray.lastIndex: Int`
`get() = size - 1`
Returns the last valid index for the array.
`public val IntArray.lastIndex: Int`
`get() = size - 1`
Returns the last valid index for the array.
`public val LongArray.lastIndex: Int`
`get() = size - 1`
Returns the last valid index for the array.
`public val FloatArray.lastIndex: Int`
`get() = size - 1`
Returns the last valid index for the array.
`public val DoubleArray.lastIndex: Int`
`get() = size - 1`
Returns the last valid index for the array.
`public val BooleanArray.lastIndex: Int`
`get() = size - 1`
Returns the last valid index for the array.
`public val CharArray.lastIndex: Int`
`get() = size - 1`
Returns an array containing all elements of the original array and then the given [element].
`@Suppress("NO_ACTUAL_FOR_EXPECT")`
`public expect operator fun <T> Array<T>.plus(element: T): Array<T>`
Returns an array containing all elements of the original array and then the given [element].
`public expect operator fun ByteArray.plus(element: Byte): ByteArray`
Returns an array containing all elements of the original array and then the given [element].
`public expect operator fun ShortArray.plus(element: Short): ShortArray`
Returns an array containing all elements of the original array and then the given [element].
`public expect operator fun IntArray.plus(element: Int): IntArray`
Returns an array containing all elements of the original array and then the given [element].
`public expect`

operator fun LongArray.plus(element: Long): LongArray\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\npublic expect operator fun FloatArray.plus(element: Float): FloatArray\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\npublic expect operator fun DoubleArray.plus(element: Double): DoubleArray\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\npublic expect operator fun BooleanArray.plus(element: Boolean): BooleanArray\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\npublic expect operator fun CharArray.plus(element: Char): CharArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\n@Suppress("NO_ACTUAL_FOR_EXPECT")\npublic expect operator fun <T> Array<T>.plus(elements: Collection<T>): Array<T>\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic expect operator fun ByteArray.plus(elements: Collection<Byte>): ByteArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic expect operator fun ShortArray.plus(elements: Collection<Short>): ShortArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic expect operator fun IntArray.plus(elements: Collection<Int>): IntArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic expect operator fun LongArray.plus(elements: Collection<Long>): LongArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic expect operator fun FloatArray.plus(elements: Collection<Float>): FloatArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic expect operator fun DoubleArray.plus(elements: Collection<Double>): DoubleArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic expect operator fun BooleanArray.plus(elements: Collection<Boolean>): BooleanArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic expect operator fun CharArray.plus(elements: Collection<Char>): CharArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress("NO_ACTUAL_FOR_EXPECT")\npublic expect operator fun <T> Array<T>.plus(elements: Array<out T>): Array<T>\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun ByteArray.plus(elements: ByteArray): ByteArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun ShortArray.plus(elements: ShortArray): ShortArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun IntArray.plus(elements: IntArray): IntArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun LongArray.plus(elements: LongArray): LongArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun FloatArray.plus(elements: FloatArray): FloatArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun DoubleArray.plus(elements: DoubleArray): DoubleArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun BooleanArray.plus(elements: BooleanArray): BooleanArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun CharArray.plus(elements: CharArray): CharArray\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n@Suppress("NO_ACTUAL_FOR_EXPECT")\npublic expect fun <T> Array<T>.plusElement(element: T): Array<T>\n\n/**\n * Sorts the array in-place.\n */\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun IntArray.sort(): Unit\n\n/**\n * Sorts the array in-place.\n */\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun LongArray.sort():

Unit\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n * \n\npublic expect fun ByteArray.sort(): Unit\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n * \n\npublic expect fun ShortArray.sort(): Unit\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n * \n\npublic expect fun DoubleArray.sort(): Unit\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n * \n\npublic expect fun FloatArray.sort(): Unit\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n * \n\npublic expect fun CharArray.sort(): Unit\n\n/**\n * Sorts the array in-place according to the natural order of its elements.\n * \n * The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n * \n * @sample samples.collections.Arrays.Sorting.sortArrayOfComparable\n * \n\npublic expect fun <T : Comparable<T>> Array<out T>.sort(): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeArrayOfComparable\n * \n\n@SinceKotlin("1.4")\n\npublic expect fun <T : Comparable<T>> Array<out T>.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n * \n\n@SinceKotlin("1.4")\n\npublic expect fun ByteArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n * \n\n@SinceKotlin("1.4")\n\npublic expect fun ShortArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n * \n\n@SinceKotlin("1.4")\n\npublic expect fun IntArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n * \n\n@SinceKotlin("1.4")\n\npublic expect fun LongArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n * \n\n@SinceKotlin("1.4")\n\npublic expect fun FloatArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n * \n\n@SinceKotlin("1.4")\n\npublic expect fun

\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

```

    */\n@SinceKotlin("1.4")\npublic fun DoubleArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n
    sort(fromIndex, toIndex)\n    reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
  
```

```

    */\n@SinceKotlin("1.4")\npublic fun CharArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n
    sort(fromIndex, toIndex)\n    reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts the array in-place according to the order specified by the given [comparator].\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n
  
```

```

    *\npublic expect fun <T> Array<out T>.sortWith(comparator: Comparator<in T>): Unit\n\n/**\n * Sorts a range in the array in-place with the given [comparator].\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
  
```

```

    *\npublic expect fun <T> Array<out T>.sortWith(comparator: Comparator<in T>, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Returns an array of Boolean containing all of the elements of this generic array.\n
  
```

```

    *\npublic fun Array<out Boolean>.toBooleanArray(): BooleanArray {\n
    return BooleanArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Byte containing all of the elements of this generic array.\n
  
```

```

    *\npublic fun Array<out Byte>.toByteArray(): ByteArray {\n
    return ByteArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Char containing all of the elements of this generic array.\n
  
```

```

    *\npublic fun Array<out Char>.toCharArray(): CharArray {\n
    return CharArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Double containing all of the elements of this generic array.\n
  
```

```

    *\npublic fun Array<out Double>.toDoubleArray(): DoubleArray {\n
    return DoubleArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Float containing all of the elements of this generic array.\n
  
```

```

    *\npublic fun Array<out Float>.toFloatArray(): FloatArray {\n
    return FloatArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Int containing all of the elements of this generic array.\n
  
```

```

    *\npublic fun Array<out Int>.toIntArray(): IntArray {\n
    return IntArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Long containing all of the elements of this generic array.\n
  
```

```

    *\npublic fun Array<out Long>.toLongArray(): LongArray {\n
    return LongArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Short containing all of the elements of this generic array.\n
  
```

```

    *\npublic fun Array<out Short>.toShortArray(): ShortArray {\n
    return ShortArray(size) { index -> this[index] }\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n
  
```

```

    *\npublic expect fun ByteArray.toTypedArray(): Array<Byte>\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n
  
```

```

    *\npublic expect fun ShortArray.toTypedArray(): Array<Short>\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n
  
```

```

    *\npublic expect fun IntArray.toTypedArray(): Array<Int>\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n
  
```

```

    *\npublic expect fun LongArray.toTypedArray(): Array<Long>\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n
  
```

```

    *\npublic expect fun FloatArray.toTypedArray(): Array<Float>\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n
  
```

```

    *\npublic expect fun DoubleArray.toTypedArray(): Array<Double>\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n
  
```

```

    *\npublic expect fun BooleanArray.toTypedArray(): Array<Boolean>\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n
  
```

```

    *\npublic expect fun CharArray.toTypedArray(): Array<Char>\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original
  
```

```

array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n *^\npublic inline
fun <T, K, V> Array<out T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> {\n  val capacity =
mapCapacity(size).coerceAtLeast(16)\n  return associateTo(LinkedHashMap<K, V>(capacity),
transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to
elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the
map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n *^\npublic inline fun <K, V>
ByteArray.associate(transform: (Byte) -> Pair<K, V>): Map<K, V> {\n  val capacity =
mapCapacity(size).coerceAtLeast(16)\n  return associateTo(LinkedHashMap<K, V>(capacity),
transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to
elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the
map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n *^\npublic inline fun <K, V>
ShortArray.associate(transform: (Short) -> Pair<K, V>): Map<K, V> {\n  val capacity =
mapCapacity(size).coerceAtLeast(16)\n  return associateTo(LinkedHashMap<K, V>(capacity),
transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to
elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the
map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n *^\npublic inline fun <K, V>
IntArray.associate(transform: (Int) -> Pair<K, V>): Map<K, V> {\n  val capacity =
mapCapacity(size).coerceAtLeast(16)\n  return associateTo(LinkedHashMap<K, V>(capacity),
transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to
elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the
map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n *^\npublic inline fun <K, V>
LongArray.associate(transform: (Long) -> Pair<K, V>): Map<K, V> {\n  val capacity =
mapCapacity(size).coerceAtLeast(16)\n  return associateTo(LinkedHashMap<K, V>(capacity),
transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to
elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the
map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n *^\npublic inline fun <K, V>
FloatArray.associate(transform: (Float) -> Pair<K, V>): Map<K, V> {\n  val capacity =
mapCapacity(size).coerceAtLeast(16)\n  return associateTo(LinkedHashMap<K, V>(capacity),
transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to
elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the
map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n *^\npublic inline fun <K, V>
DoubleArray.associate(transform: (Double) -> Pair<K, V>): Map<K, V> {\n  val capacity =
mapCapacity(size).coerceAtLeast(16)\n  return associateTo(LinkedHashMap<K, V>(capacity),
transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to
elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the
map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n *^\npublic inline fun <K, V>
BooleanArray.associate(transform: (Boolean) -> Pair<K, V>): Map<K, V> {\n  val capacity =
mapCapacity(size).coerceAtLeast(16)\n  return associateTo(LinkedHashMap<K, V>(capacity),
transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to
elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the
map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample

```

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n */\npublic inline fun <K, V>
CharArray.associate(transform: (Char) -> Pair<K, V>): Map<K, V> {\n    val capacity =
mapCapacity(size).coerceAtLeast(16)\n    return associateTo(LinkedHashMap<K, V>(capacity),
transform)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <T, K>
Array<out T>.associateBy(keySelector: (T) -> K): Map<K, T> {\n    val capacity =
mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, T>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K>
ByteArray.associateBy(keySelector: (Byte) -> K): Map<K, Byte> {\n    val capacity =
mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, Byte>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K>
ShortArray.associateBy(keySelector: (Short) -> K): Map<K, Short> {\n    val capacity =
mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, Short>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K>
IntArray.associateBy(keySelector: (Int) -> K): Map<K, Int> {\n    val capacity =
mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, Int>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K>
LongArray.associateBy(keySelector: (Long) -> K): Map<K, Long> {\n    val capacity =
mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, Long>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K>
FloatArray.associateBy(keySelector: (Float) -> K): Map<K, Float> {\n    val capacity =
mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, Float>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration

```

order of the original array.\n * \n * @sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K>
DoubleArray.associateBy(keySelector: (Double) -> K): Map<K, Double> {\n  val capacity =
mapCapacity(size).coerceAtLeast(16)\n  return associateByTo(LinkedHashMap<K, Double>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K>

```

BooleanArray.associateBy(keySelector: (Boolean) -> K): Map<K, Boolean> {\n  val capacity =
mapCapacity(size).coerceAtLeast(16)\n  return associateByTo(LinkedHashMap<K, Boolean>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K>

```

CharArray.associateBy(keySelector: (Char) -> K): Map<K, Char> {\n  val capacity =
mapCapacity(size).coerceAtLeast(16)\n  return associateByTo(LinkedHashMap<K, Char>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform] and indexed by
[keySelector] functions applied to elements of the given array.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n */\npublic inline

```

fun <T, K, V> Array<out T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> {\n  val
capacity = mapCapacity(size).coerceAtLeast(16)\n  return associateByTo(LinkedHashMap<K, V>(capacity),
keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform]
and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would
have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves
the entry iteration order of the original array.\n * \n * @sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n */\npublic inline

```

fun <K, V> ByteArray.associateBy(keySelector: (Byte) -> K, valueTransform: (Byte) -> V): Map<K, V> {\n  val
capacity = mapCapacity(size).coerceAtLeast(16)\n  return associateByTo(LinkedHashMap<K, V>(capacity),
keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform]
and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would
have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves
the entry iteration order of the original array.\n * \n * @sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n */\npublic inline

```

fun <K, V> ShortArray.associateBy(keySelector: (Short) -> K, valueTransform: (Short) -> V): Map<K, V> {\n
val capacity = mapCapacity(size).coerceAtLeast(16)\n  return associateByTo(LinkedHashMap<K, V>(capacity),
keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform]
and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would
have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves
the entry iteration order of the original array.\n * \n * @sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n */\npublic inline

```

fun <K, V> IntArray.associateBy(keySelector: (Int) -> K, valueTransform: (Int) -> V): Map<K, V> {\n  val
capacity = mapCapacity(size).coerceAtLeast(16)\n  return associateByTo(LinkedHashMap<K, V>(capacity),
keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform]
and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would

```

have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n * \n public inline
fun <K, V> LongArray.associateBy(keySelector: (Long) -> K, valueTransform: (Long) -> V): Map<K, V> {\n val
capacity = mapCapacity(size).coerceAtLeast(16)\n return associateByTo(LinkedHashMap<K, V>(capacity),
keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform]
and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would
have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves
the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n * \n public inline
fun <K, V> FloatArray.associateBy(keySelector: (Float) -> K, valueTransform: (Float) -> V): Map<K, V> {\n val
capacity = mapCapacity(size).coerceAtLeast(16)\n return associateByTo(LinkedHashMap<K, V>(capacity),
keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform]
and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would
have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves
the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n * \n public inline
fun <K, V> DoubleArray.associateBy(keySelector: (Double) -> K, valueTransform: (Double) -> V): Map<K, V>
{\n val capacity = mapCapacity(size).coerceAtLeast(16)\n return associateByTo(LinkedHashMap<K,
V>(capacity), keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by
[valueTransform] and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two
elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The
returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n * \n public inline
fun <K, V> BooleanArray.associateBy(keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): Map<K, V>
{\n val capacity = mapCapacity(size).coerceAtLeast(16)\n return associateByTo(LinkedHashMap<K,
V>(capacity), keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by
[valueTransform] and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two
elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The
returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n * \n public inline
fun <K, V> CharArray.associateBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, V> {\n val
capacity = mapCapacity(size).coerceAtLeast(16)\n return associateByTo(LinkedHashMap<K, V>(capacity),
keySelector, valueTransform)\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value
pairs,\n * where key is provided by the [keySelector] function applied to each element of the given array\n * and
value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last
one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n * \n public inline fun <T, K, M :
MutableMap<in K, in T>> Array<out T>.associateByTo(destination: M, keySelector: (T) -> K): M {\n for
(element in this) {\n destination.put(keySelector(element), element)\n }\n return destination\n}\n\n/**\n *
Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the
[keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any
two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n *
@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n * \n public inline fun <K,
M : MutableMap<in K, in Byte>> ByteArray.associateByTo(destination: M, keySelector: (Byte) -> K): M {\n for
(element in this) {\n destination.put(keySelector(element), element)\n }\n return destination\n}\n\n/**\n *
Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the
[keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any

```


two elements would have the same key returned by [keySelector] the last one gets added to the map.

```

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo
public inline fun <K,
M : MutableMap<in K, in Short>> ShortArray.associateByTo(destination: M, keySelector: (Short) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)
    }
    return destination
}

```

Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function applied to each element of the given array and value is the element itself. If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

```

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo
public inline fun <K,
M : MutableMap<in K, in Int>> IntArray.associateByTo(destination: M, keySelector: (Int) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)
    }
    return destination
}

```

Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function applied to each element of the given array and value is the element itself. If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

```

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo
public inline fun <K,
M : MutableMap<in K, in Long>> LongArray.associateByTo(destination: M, keySelector: (Long) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)
    }
    return destination
}

```

Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function applied to each element of the given array and value is the element itself. If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

```

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo
public inline fun <K,
M : MutableMap<in K, in Float>> FloatArray.associateByTo(destination: M, keySelector: (Float) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)
    }
    return destination
}

```

Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function applied to each element of the given array and value is the element itself. If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

```

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo
public inline fun <K,
M : MutableMap<in K, in Double>> DoubleArray.associateByTo(destination: M, keySelector: (Double) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)
    }
    return destination
}

```

Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function applied to each element of the given array and value is the element itself. If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

```

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo
public inline fun <K, M : MutableMap<in K, in Boolean>> BooleanArray.associateByTo(destination: M, keySelector:
(Boolean) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)
    }
    return destination
}

```

Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function applied to each element of the given array and value is the element itself. If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

```

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo
public inline fun <K, M : MutableMap<in K, in Char>> CharArray.associateByTo(destination: M, keySelector: (Char) ->
K): M {
    for (element in this) {
        destination.put(keySelector(element), element)
    }
    return destination
}

```

Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function and value is provided by the [valueTransform] function applied to elements of the given array. If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

```

@sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform
public inline fun <T, K, V, M : MutableMap<in K, in V>> Array<out T>.associateByTo(destination: M, keySelector: (T) -
> K, valueTransform: (T) -> V): M {
    for (element in this) {
        destination.put(keySelector(element),
valueTransform(element))
    }
    return destination
}

```

Populates and returns the [destination] mutable

map with key-value pairs, where key is provided by the [keySelector] function and value is provided by the [valueTransform] function applied to elements of the given array. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. @sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform
public inline fun <K, V, M : MutableMap<in K, in V>> ByteArray.associateByTo(destination: M, keySelector: (Byte) -> K, valueTransform: (Byte) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}

```

Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function and value is provided by the [valueTransform] function applied to elements of the given array. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. @sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform
public inline fun <K, V, M : MutableMap<in K, in V>> ShortArray.associateByTo(destination: M, keySelector: (Short) -> K, valueTransform: (Short) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}

```

Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function and value is provided by the [valueTransform] function applied to elements of the given array. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. @sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform
public inline fun <K, V, M : MutableMap<in K, in V>> IntArray.associateByTo(destination: M, keySelector: (Int) -> K, valueTransform: (Int) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}

```

Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function and value is provided by the [valueTransform] function applied to elements of the given array. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. @sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform
public inline fun <K, V, M : MutableMap<in K, in V>> LongArray.associateByTo(destination: M, keySelector: (Long) -> K, valueTransform: (Long) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}

```

Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function and value is provided by the [valueTransform] function applied to elements of the given array. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. @sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform
public inline fun <K, V, M : MutableMap<in K, in V>> FloatArray.associateByTo(destination: M, keySelector: (Float) -> K, valueTransform: (Float) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}

```

Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function and value is provided by the [valueTransform] function applied to elements of the given array. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. @sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform
public inline fun <K, V, M : MutableMap<in K, in V>> DoubleArray.associateByTo(destination: M, keySelector: (Double) -> K, valueTransform: (Double) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}

```

Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function and value is provided by the [valueTransform] function applied to elements of the given array. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. @sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform
public inline fun <K, V, M : MutableMap<in K, in V>> BooleanArray.associateByTo(destination: M, keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): M {
    for (element in this) {

```

```

destination.put(keySelector(element), valueTransform(element))\n } \n return destination\n}\n\n/**\n * Populates
and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector]
function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n *
\n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n *
\n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n
*/\npublic inline fun <K, V, M : MutableMap<in K, in V>> CharArray.associateByTo(destination: M, keySelector:
(Char) -> K, valueTransform: (Char) -> V): M {\n for (element in this) {\n
destination.put(keySelector(element), valueTransform(element))\n } \n return destination\n}\n\n/**\n * Populates
and returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each
element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the
map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic
inline fun <T, K, V, M : MutableMap<in K, in V>> Array<out T>.associateTo(destination: M, transform: (T) ->
Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n } \n return
destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by
[transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key
the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic inline fun <K, V, M :
MutableMap<in K, in V>> ByteArray.associateTo(destination: M, transform: (Byte) -> Pair<K, V>): M {\n for
(element in this) {\n destination += transform(element)\n } \n return destination\n}\n\n/**\n * Populates and
returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each
element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the
map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic
inline fun <K, V, M : MutableMap<in K, in V>> ShortArray.associateTo(destination: M, transform: (Short) ->
Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n } \n return
destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by
[transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key
the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic inline fun <K, V, M :
MutableMap<in K, in V>> IntArray.associateTo(destination: M, transform: (Int) -> Pair<K, V>): M {\n for
(element in this) {\n destination += transform(element)\n } \n return destination\n}\n\n/**\n * Populates and
returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each
element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the
map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic
inline fun <K, V, M : MutableMap<in K, in V>> LongArray.associateTo(destination: M, transform: (Long) ->
Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n } \n return
destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by
[transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key
the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic inline fun <K, V, M :
MutableMap<in K, in V>> FloatArray.associateTo(destination: M, transform: (Float) -> Pair<K, V>): M {\n for
(element in this) {\n destination += transform(element)\n } \n return destination\n}\n\n/**\n * Populates and
returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each
element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the
map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic
inline fun <K, V, M : MutableMap<in K, in V>> DoubleArray.associateTo(destination: M, transform: (Double) ->
Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n } \n return
destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by
[transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key
the last one gets added to the map.\n * \n * @sample

```

the last one gets added to the map.

```
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n\npublic inline fun <K, V, M : MutableMap<in K, in V>> BooleanArray.associateTo(destination: M, transform: (Boolean) -> Pair<K, V>): M {\n    for (element in this) {\n        destination += transform(element)\n    }\n    return destination\n}\n\nPopulates and returns the [destination] mutable map with key-value pairs provided by [transform] function applied to each element of the given array.\n\nIf any of two pairs would have the same key the last one gets added to the map.\n\n@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n\npublic inline fun <K, V, M : MutableMap<in K, in V>> CharArray.associateTo(destination: M, transform: (Char) -> Pair<K, V>): M {\n    for (element in this) {\n        destination += transform(element)\n    }\n    return destination\n}\n\nReturns a [Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element.\n\nIf any two elements are equal, the last one gets added to the map.\n\nThe returned map preserves the entry iteration order of the original array.\n\n@sample samples.collections.Collections.Transformations.associateWith\n\n@SinceKotlin("1.4")\n\npublic inline fun <K, V> Array<out K>.associateWith(valueSelector: (K) -> V): Map<K, V> {\n    val result = LinkedHashMap<K, V>(mapCapacity(size).coerceAtLeast(16))\n    return associateWithTo(result, valueSelector)\n}\n\nReturns a [Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element.\n\nIf any two elements are equal, the last one gets added to the map.\n\nThe returned map preserves the entry iteration order of the original array.\n\n@sample samples.collections.Collections.Transformations.associateWith\n\n@SinceKotlin("1.4")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <V> ByteArray.associateWith(valueSelector: (Byte) -> V): Map<Byte, V> {\n    val result = LinkedHashMap<Byte, V>(mapCapacity(size).coerceAtLeast(16))\n    return associateWithTo(result, valueSelector)\n}\n\nReturns a [Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element.\n\nIf any two elements are equal, the last one gets added to the map.\n\nThe returned map preserves the entry iteration order of the original array.\n\n@sample samples.collections.Collections.Transformations.associateWith\n\n@SinceKotlin("1.4")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <V> ShortArray.associateWith(valueSelector: (Short) -> V): Map<Short, V> {\n    val result = LinkedHashMap<Short, V>(mapCapacity(size).coerceAtLeast(16))\n    return associateWithTo(result, valueSelector)\n}\n\nReturns a [Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element.\n\nIf any two elements are equal, the last one gets added to the map.\n\nThe returned map preserves the entry iteration order of the original array.\n\n@sample samples.collections.Collections.Transformations.associateWith\n\n@SinceKotlin("1.4")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <V> IntArray.associateWith(valueSelector: (Int) -> V): Map<Int, V> {\n    val result = LinkedHashMap<Int, V>(mapCapacity(size).coerceAtLeast(16))\n    return associateWithTo(result, valueSelector)\n}\n\nReturns a [Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element.\n\nIf any two elements are equal, the last one gets added to the map.\n\nThe returned map preserves the entry iteration order of the original array.\n\n@sample samples.collections.Collections.Transformations.associateWith\n\n@SinceKotlin("1.4")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <V> LongArray.associateWith(valueSelector: (Long) -> V): Map<Long, V> {\n    val result = LinkedHashMap<Long, V>(mapCapacity(size).coerceAtLeast(16))\n    return associateWithTo(result, valueSelector)\n}\n\nReturns a [Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element.\n\nIf any two elements are equal, the last one gets added to the map.\n\nThe returned map preserves the entry iteration order of the original array.\n\n@sample samples.collections.Collections.Transformations.associateWith\n\n@SinceKotlin("1.4")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <V>
```

```

FloatArray.associateWith(valueSelector: (Float) -> V): Map<Float, V> {\n    val result = LinkedHashMap<Float,
V>(mapCapacity(size).coerceAtLeast(16))\n    return associateWithTo(result, valueSelector)\n}\n\n/**\n * Returns a
[Map] where keys are elements from the given array and values are\n * produced by the [valueSelector] function
applied to each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The
returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Collections.Transformations.associateWith\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V>
DoubleArray.associateWith(valueSelector: (Double) -> V): Map<Double, V> {\n    val result =
LinkedHashMap<Double, V>(mapCapacity(size).coerceAtLeast(16))\n    return associateWithTo(result,
valueSelector)\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n *
produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the last one
gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n *
@sample samples.collections.Collections.Transformations.associateWith\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V>
BooleanArray.associateWith(valueSelector: (Boolean) -> V): Map<Boolean, V> {\n    val result =
LinkedHashMap<Boolean, V>(mapCapacity(size).coerceAtLeast(16))\n    return associateWithTo(result,
valueSelector)\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n *
produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the last one
gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n *
@sample samples.collections.Collections.Transformations.associateWith\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V>
CharArray.associateWith(valueSelector: (Char) -> V): Map<Char, V> {\n    val result = LinkedHashMap<Char,
V>(mapCapacity(size).coerceAtMost(128)).coerceAtLeast(16))\n    return associateWithTo(result,
valueSelector)\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each
element of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function
applied to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n
* @sample samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, M : MutableMap<in K, in V>>
Array<out K>.associateWithTo(destination: M, valueSelector: (K)
-> V): M {\n    for (element in this) {\n        destination.put(element, valueSelector(element))\n    }\n    return
destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each element
of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function applied
to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n *
@sample samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Byte, in V>>
ByteArray.associateWithTo(destination: M, valueSelector: (Byte) -> V): M {\n    for (element in this) {\n
destination.put(element, valueSelector(element))\n    }\n    return destination\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element
itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are
equal, the last one overwrites the former value in the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Short, in V>>
ShortArray.associateWithTo(destination: M, valueSelector: (Short) -> V): M {\n    for (element in this) {\n
destination.put(element, valueSelector(element))\n    }\n    return destination\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element
itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are
equal, the last one overwrites the former value in the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Int, in V>>

```

```

IntArray.associateWithTo(destination: M, valueSelector: (Int) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

Populates and returns the [destination] mutable map with key-value pairs for each element of the given array, where key is the element itself and value is provided by the [valueSelector] function applied to that key. If any two elements are equal, the last one overwrites the former value in the map.

@sample
samples.collections.Collections.Transformations.associateWithTo

*SinceKotlin("1.4")@kotlin.internal.InlineOnly
public inline fun <V, M : MutableMap<in Long, in V>>
LongArray.associateWithTo(destination: M, valueSelector: (Long) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

Populates and returns the [destination] mutable map with key-value pairs for each element of the given array, where key is the element itself and value is provided by the [valueSelector] function applied to that key. If any two elements are equal, the last one overwrites the former value in the map.

@sample
samples.collections.Collections.Transformations.associateWithTo

*SinceKotlin("1.4")@kotlin.internal.InlineOnly
public inline fun <V, M : MutableMap<in Float, in V>>
FloatArray.associateWithTo(destination: M, valueSelector: (Float) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

Populates and returns the [destination] mutable map with key-value pairs for each element of the given array, where key is the element itself and value is provided by the [valueSelector] function applied to that key. If any two elements are equal, the last one overwrites the former value in the map.

@sample
samples.collections.Collections.Transformations.associateWithTo

*SinceKotlin("1.4")@kotlin.internal.InlineOnly
public inline fun <V, M : MutableMap<in Double, in V>>
DoubleArray.associateWithTo(destination: M, valueSelector: (Double) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

Populates and returns the [destination] mutable map with key-value pairs for each element of the given array, where key is the element itself and value is provided by the [valueSelector] function applied to that key. If any two elements are equal, the last one overwrites the former value in the map.

@sample
samples.collections.Collections.Transformations.associateWithTo

*SinceKotlin("1.4")@kotlin.internal.InlineOnly
public inline fun <V, M : MutableMap<in Boolean, in V>>
BooleanArray.associateWithTo(destination: M, valueSelector: (Boolean) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

Populates and returns the [destination] mutable map with key-value pairs for each element of the given array, where key is the element itself and value is provided by the [valueSelector] function applied to that key. If any two elements are equal, the last one overwrites the former value in the map.

@sample
samples.collections.Collections.Transformations.associateWithTo

*SinceKotlin("1.4")@kotlin.internal.InlineOnly
public inline fun <V, M : MutableMap<in Char, in V>>
CharArray.associateWithTo(destination: M, valueSelector: (Char) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

Appends all elements to the given [destination] collection.

public fun <T, C : MutableCollection<in T>> Array<out T>.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

Appends all elements to the given [destination] collection.

public fun <C : MutableCollection<in Byte>> ByteArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

Appends all elements to the given [destination] collection.

public fun <C : MutableCollection<in Short>> ShortArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

Appends all elements to the given [destination] collection.

public fun <C : MutableCollection<in Int>> IntArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

Appends all elements to the given [destination] collection.

public fun <C : MutableCollection<in Long>>
LongArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return

```

```

destination\n}\n\n/**\n * Appends all elements to the given [destination] collection.\n */\npublic fun <C :
MutableCollection<in Float>> FloatArray.toCollection(destination: C): C {\n  for (item in this) {\n
destination.add(item)\n  }\n  return destination\n}\n\n/**\n * Appends all elements to the given [destination]
collection.\n */\npublic fun <C : MutableCollection<in Double>> DoubleArray.toCollection(destination: C): C {\n
for (item in this) {\n  destination.add(item)\n  }\n  return destination\n}\n\n/**\n * Appends all elements to the
given [destination] collection.\n */\npublic fun <C : MutableCollection<in Boolean>>
BooleanArray.toCollection(destination: C): C {\n  for (item in this) {\n  destination.add(item)\n  }\n  return
destination\n}\n\n/**\n * Appends all elements to the given [destination] collection.\n */\npublic fun <C :
MutableCollection<in Char>> CharArray.toCollection(destination: C): C {\n  for (item in this) {\n
destination.add(item)\n  }\n  return destination\n}\n\n/**\n * Returns a new [HashSet] of all elements.\n
*/\npublic fun <T> Array<out T>.toHashSet(): HashSet<T> {\n  return
toCollection(HashSet<T>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [HashSet] of all elements.\n
*/\npublic fun ByteArray.toHashSet(): HashSet<Byte> {\n  return
toCollection(HashSet<Byte>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [HashSet] of all elements.\n
*/\npublic fun ShortArray.toHashSet(): HashSet<Short> {\n  return
toCollection(HashSet<Short>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [HashSet] of all elements.\n
*/\npublic fun IntArray.toHashSet(): HashSet<Int> {\n  return
toCollection(HashSet<Int>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [HashSet] of all elements.\n
*/\npublic fun LongArray.toHashSet(): HashSet<Long> {\n  return
toCollection(HashSet<Long>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [HashSet] of all elements.\n
*/\npublic fun FloatArray.toHashSet(): HashSet<Float> {\n  return
toCollection(HashSet<Float>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [HashSet] of all elements.\n
*/\npublic fun DoubleArray.toHashSet(): HashSet<Double> {\n  return
toCollection(HashSet<Double>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [HashSet] of all elements.\n
*/\npublic fun BooleanArray.toHashSet(): HashSet<Boolean> {\n  return
toCollection(HashSet<Boolean>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [HashSet] of all elements.\n
*/\npublic fun CharArray.toHashSet(): HashSet<Char> {\n  return
toCollection(HashSet<Char>(mapCapacity(size.coerceAtMost(128))))\n}\n\n/**\n * Returns a [List] containing all
elements.\n */\npublic fun <T> Array<out T>.toList(): List<T> {\n  return when (size) {\n    0 -> emptyList()\n
1 -> listOf(this[0])\n    else -> this.toMutableList()\n  }\n}\n\n/**\n * Returns a [List] containing all
elements.\n */\npublic fun ByteArray.toList(): List<Byte> {\n  return when (size) {\n    0 -> emptyList()\n
1 -> listOf(this[0])\n    else -> this.toMutableList()\n  }\n}\n\n/**\n * Returns a [List] containing all elements.\n
*/\npublic fun ShortArray.toList(): List<Short> {\n  return when (size) {\n    0 -> emptyList()\n
1 -> listOf(this[0])\n    else -> this.toMutableList()\n  }\n}\n\n/**\n * Returns a [List] containing all elements.\n
*/\npublic fun IntArray.toList(): List<Int> {\n  return when (size) {\n    0 -> emptyList()\n
1 -> listOf(this[0])\n    else -> this.toMutableList()\n  }\n}\n\n/**\n * Returns a [List] containing all elements.\n
*/\npublic fun LongArray.toList(): List<Long> {\n  return when (size) {\n    0 -> emptyList()\n
1 -> listOf(this[0])\n    else -> this.toMutableList()\n  }\n}\n\n/**\n * Returns a [List] containing all elements.\n
*/\npublic fun FloatArray.toList(): List<Float> {\n  return when (size) {\n    0 -> emptyList()\n
1 -> listOf(this[0])\n    else -> this.toMutableList()\n  }\n}\n\n/**\n * Returns a [List] containing all elements.\n
*/\npublic fun DoubleArray.toList(): List<Double> {\n  return when (size) {\n    0 -> emptyList()\n
1 -> listOf(this[0])\n    else -> this.toMutableList()\n  }\n}\n\n/**\n * Returns a [List] containing all elements.\n
*/\npublic fun BooleanArray.toList(): List<Boolean> {\n  return when (size) {\n    0 -> emptyList()\n
1 -> listOf(this[0])\n    else -> this.toMutableList()\n  }\n}\n\n/**\n * Returns a [List] containing all elements.\n
*/\npublic fun CharArray.toList(): List<Char> {\n  return when (size) {\n    0 -> emptyList()\n
1 -> listOf(this[0])\n    else -> this.toMutableList()\n  }\n}\n\n/**\n * Returns a new [MutableList] filled with all
elements of this array.\n */\npublic fun <T> Array<out T>.toMutableList(): MutableList<T> {\n  return
ArrayList(this.asCollection())\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n

```

```

*\npublic fun ByteArray.toMutableList(): MutableList<Byte> {\n    val list = ArrayList<Byte>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun ShortArray.toMutableList(): MutableList<Short> {\n    val list = ArrayList<Short>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun IntArray.toMutableList(): MutableList<Int> {\n    val list = ArrayList<Int>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun LongArray.toMutableList(): MutableList<Long> {\n    val list = ArrayList<Long>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun FloatArray.toMutableList(): MutableList<Float> {\n    val list = ArrayList<Float>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun DoubleArray.toMutableList(): MutableList<Double> {\n    val list = ArrayList<Double>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun BooleanArray.toMutableList(): MutableList<Boolean> {\n    val list = ArrayList<Boolean>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun CharArray.toMutableList(): MutableList<Char> {\n    val list = ArrayList<Char>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun <T> Array<out T>.toSet(): Set<T> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<T>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun ByteArray.toSet(): Set<Byte> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Byte>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun ShortArray.toSet(): Set<Short> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Short>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun IntArray.toSet(): Set<Int> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Int>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun LongArray.toSet(): Set<Long> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Long>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun FloatArray.toSet(): Set<Float> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Float>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun DoubleArray.toSet(): Set<Double> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Double>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun BooleanArray.toSet(): Set<Boolean> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Boolean>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun CharArray.toSet(): Set<Char> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Char>(mapCapacity(size.coerceAtMost(128))))\n    }\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <T, R> Array<out T>.flatMap(transform: (T) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being invoked on each element of

```



```

original array.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n * \npublic inline fun
<R> ByteArray.flatMap(transform: (Byte) -> Iterable<R>): List<R> {\n  return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n * \npublic inline fun <R> ShortArray.flatMap(transform:
(Short) -> Iterable<R>): List<R> {\n  return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single
list of all elements yielded from results of [transform] function being invoked on each element of original array.\n *
\n * @sample samples.collections.Collections.Transformations.flatMap\n * \npublic inline fun <R>
IntArray.flatMap(transform: (Int) -> Iterable<R>): List<R> {\n  return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n * \npublic inline fun <R> LongArray.flatMap(transform:
(Long) -> Iterable<R>): List<R> {\n  return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single
list of all elements yielded from results of [transform] function being invoked on each element of original array.\n *
\n * @sample samples.collections.Collections.Transformations.flatMap\n * \npublic inline fun <R>
FloatArray.flatMap(transform: (Float) -> Iterable<R>): List<R> {\n  return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n * \npublic inline fun <R>
DoubleArray.flatMap(transform: (Double) -> Iterable<R>): List<R> {\n  return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n * \npublic inline fun <R>
BooleanArray.flatMap(transform: (Boolean) -> Iterable<R>): List<R> {\n  return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n * \npublic inline fun <R> CharArray.flatMap(transform:
(Char) -> Iterable<R>): List<R> {\n  return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single
list of all elements yielded from results of [transform] function being invoked on each element of original array.\n *
\n * @sample samples.collections.Collections.Transformations.flatMap\n
*\n * \n@SinceKotlin("1.4")\n * \n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n * \n@OverloadResolution
ByLambdaReturnType\n * \n@kotlin.jvm.JvmName("flatMapSequence")\n * \npublic inline fun <T, R> Array<out
T>.flatMap(transform: (T) -> Sequence<R>): List<R> {\n  return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element\n * and its index in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMapIndexed\n
*\n * \n@SinceKotlin("1.4")\n * \n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n * \n@OverloadResolution
ByLambdaReturnType\n * \n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n * \n@kotlin.internal.InlineOnly\n * \npublic
inline fun <T, R> Array<out T>.flatMapIndexed(transform: (index: Int, T) -> Iterable<R>): List<R> {\n  return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
\n * @sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n * \n@SinceKotlin("1.4")\n * \n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n * \n@OverloadResolution
ByLambdaReturnType\n * \n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n * \n@kotlin.internal.InlineOnly\n * \npublic
inline fun <R> ByteArray.flatMapIndexed(transform: (index: Int, Byte) -> Iterable<R>): List<R> {\n  return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
\n * @sample samples.collections.Collections.Transformations.flatMapIndexed\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> ShortArray.flatMapIndexed(transform: (index: Int, Short) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> IntArray.flatMapIndexed(transform: (index: Int, Int) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> LongArray.flatMapIndexed(transform: (index: Int, Long) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> FloatArray.flatMapIndexed(transform: (index: Int, Float) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> DoubleArray.flatMapIndexed(transform: (index: Int, Double) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> BooleanArray.flatMapIndexed(transform: (index: Int, Boolean) -> Iterable<R>): List<R> {\n
return flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded
from results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> CharArray.flatMapIndexed(transform: (index: Int, Char) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedSequence")\n@kotlin.internal.InlineOnly\npubli
c inline fun <T, R> Array<out T>.flatMapIndexed(transform: (index: Int, T) -> Sequence<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n

```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapIndexedTo(destination: C, transform:\n(index: Int, T) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(index++,\n        element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from\n results of [transform] function being invoked on each element\n * and its index in the original array, to the given\n [destination].\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> ByteArray.flatMapIndexedTo(destination: C, transform: (index: Int,\nByte) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(index++,\n        element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from\n results of [transform] function being invoked on each element\n * and its index in the original array, to the given\n [destination].\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> ShortArray.flatMapIndexedTo(destination: C, transform: (index: Int,\nShort) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(index++,\n        element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from\n results of [transform] function being invoked on each element\n * and its index in the original array, to the given\n [destination].\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> IntArray.flatMapIndexedTo(destination: C, transform: (index: Int,\nInt) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(index++, element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from results of\n [transform] function being invoked on each element\n * and its index in the original array, to the given\n [destination].\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> LongArray.flatMapIndexedTo(destination: C, transform: (index: Int,\nLong) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(index++,\n        element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from\n results of [transform] function being invoked on each element\n * and its index in the original array, to the given\n [destination].\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> FloatArray.flatMapIndexedTo(destination: C, transform: (index: Int,\nFloat) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(index++,\n        element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from\n results of [transform] function being invoked on each element\n * and its index in the original array, to the given\n [destination].\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> DoubleArray.flatMapIndexedTo(destination: C, transform: (index:\nInt, Double) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(index++,\n        element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from\n results of [transform] function being invoked on each element\n * and its index in the original array, to the given
```

```

[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <R, C : MutableCollection<in R>> BooleanArray.flatMapIndexedTo(destination: C, transform: (index:
Int, Boolean) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(index++,
element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <R, C : MutableCollection<in R>> CharArray.flatMapIndexedTo(destination: C, transform: (index: Int,
Char) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(index++,
element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapIndexedSequenceTo")\n@kotlin.internal.InlineOnly\npu
blic inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapIndexedTo(destination: C, transform:
(index: Int, T) -> Sequence<R>): C {\n    var index = 0\n    for (element in this) {\n        val list =
transform(index++, element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all
elements yielded from results of [transform] function being invoked on each element of original array, to the given
[destination].\n *\npublic inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapTo(destination: C,
transform: (T) -> Iterable<R>): C {\n    for (element in this) {\n        val list = transform(element)\n
destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element of original array, to the given [destination].\n *\npublic inline
fun <R, C : MutableCollection<in R>> ByteArray.flatMapTo(destination: C, transform: (Byte) -> Iterable<R>): C
{\n    for (element in this) {\n        val list = transform(element)\n        destination.addAll(list)\n    }\n    return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original array, to the given [destination].\n *\npublic inline fun <R, C : MutableCollection<in R>>
ShortArray.flatMapTo(destination: C, transform: (Short) -> Iterable<R>): C {\n    for (element in this) {\n        val
list = transform(element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all
elements yielded from results of [transform] function being invoked on each element of original array, to the given
[destination].\n *\npublic inline fun <R, C : MutableCollection<in R>> IntArray.flatMapTo(destination: C,
transform: (Int) -> Iterable<R>): C {\n    for (element in this) {\n        val list = transform(element)\n
destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element of original array, to the given [destination].\n *\npublic inline
fun <R, C : MutableCollection<in R>> LongArray.flatMapTo(destination: C, transform: (Long) -> Iterable<R>): C
{\n    for (element in this) {\n        val list = transform(element)\n        destination.addAll(list)\n    }\n    return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original array, to the given [destination].\n *\npublic inline fun <R, C : MutableCollection<in R>>
FloatArray.flatMapTo(destination: C, transform: (Float) -> Iterable<R>): C {\n    for (element in this) {\n        val
list = transform(element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all
elements yielded from results of [transform] function being invoked on each element of original array, to the given
[destination].\n *\npublic inline fun <R, C : MutableCollection<in R>> DoubleArray.flatMapTo(destination: C,
transform: (Double) -> Iterable<R>): C {\n    for (element in this) {\n        val list = transform(element)\n
destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element of original array, to the given [destination].\n *\npublic inline
fun <R, C : MutableCollection<in R>> BooleanArray.flatMapTo(destination: C, transform: (Boolean) ->

```



```

inline fun <K> DoubleArray.groupBy(keySelector: (Double) -> K): Map<K, List<Double>> {
    return groupByTo(LinkedHashMap<K, MutableList<Double>>(), keySelector)
}
/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and returns a map where each group key is associated with a list of
 * corresponding elements.
 *
 * The returned map preserves the entry iteration order of the keys produced from the original
 * array.
 *
 * @sample samples.collections.Collections.Transformations.groupBy
 */
public inline fun <K> BooleanArray.groupBy(keySelector: (Boolean) -> K): Map<K, List<Boolean>> {
    return groupByTo(LinkedHashMap<K, MutableList<Boolean>>(), keySelector)
}
/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and returns a map where each group key is associated with a list of
 * corresponding elements.
 *
 * The returned map preserves the entry iteration order of the keys produced from the original
 * array.
 *
 * @sample samples.collections.Collections.Transformations.groupBy
 */
public inline fun <K> CharArray.groupBy(keySelector: (Char) -> K): Map<K, List<Char>> {
    return groupByTo(LinkedHashMap<K, MutableList<Char>>(), keySelector)
}
/**
 * Groups values returned by the [valueTransform] function applied to each element of the
 * original array
 * by the key returned by the given [keySelector] function applied to the element
 * and returns a map where each group key is associated with a list of corresponding values.
 *
 * The returned map preserves the entry iteration order of the keys produced from the original
 * array.
 *
 * @sample samples.collections.Collections.Transformations.groupByKeysAndValues
 */
public inline fun <T, K, V> Array<out T>.groupBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}
/**
 * Groups values returned by the [valueTransform] function applied to each element of the
 * original array
 * by the key returned by the given [keySelector] function applied to the element
 * and returns a map where each group key is associated with a list of corresponding values.
 *
 * The returned map preserves the entry iteration order of the keys produced from the original
 * array.
 *
 * @sample samples.collections.Collections.Transformations.groupByKeysAndValues
 */
public inline fun <K, V> ByteArray.groupBy(keySelector: (Byte) -> K, valueTransform: (Byte) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}
/**
 * Groups values returned by the [valueTransform] function applied to each element of the
 * original array
 * by the key returned by the given [keySelector] function applied to the element
 * and returns a map where each group key is associated with a list of corresponding values.
 *
 * The returned map preserves the entry iteration order of the keys produced from the original
 * array.
 *
 * @sample samples.collections.Collections.Transformations.groupByKeysAndValues
 */
public inline fun <K, V> ShortArray.groupBy(keySelector: (Short) -> K, valueTransform: (Short) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}
/**
 * Groups values returned by the [valueTransform] function applied to each element of the
 * original array
 * by the key returned by the given [keySelector] function applied to the element
 * and returns a map where each group key is associated with a list of corresponding values.
 *
 * The returned map preserves the entry iteration order of the keys produced from the original
 * array.
 *
 * @sample samples.collections.Collections.Transformations.groupByKeysAndValues
 */
public inline fun <K, V> IntArray.groupBy(keySelector: (Int) -> K, valueTransform: (Int) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}
/**
 * Groups values returned by the [valueTransform] function applied to each element of the
 * original array
 * by the key returned by the given [keySelector] function applied to the element
 * and returns a map where each group key is associated with a list of corresponding values.
 *
 * The returned map preserves the entry iteration order of the keys produced from the original
 * array.
 *
 * @sample samples.collections.Collections.Transformations.groupByKeysAndValues
 */
public inline fun <K, V> LongArray.groupBy(keySelector: (Long) -> K, valueTransform: (Long) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}

```

returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample

```

samples.collections.Collections.Transformations.groupByKeyAndValues\n */\npublic inline fun <K, V>
FloatArray.groupBy(keySelector: (Float) -> K, valueTransform: (Float) -> V): Map<K, List<V>> {\n  return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n }\n\n/**\n * Groups values
returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n */\npublic inline fun <K, V>
DoubleArray.groupBy(keySelector: (Double) -> K, valueTransform: (Double) -> V): Map<K, List<V>> {\n
return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n }\n\n/**\n * Groups
values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n */\npublic inline fun <K, V>
BooleanArray.groupBy(keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): Map<K, List<V>> {\n
return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n }\n\n/**\n * Groups
values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n */\npublic inline fun <K, V>
CharArray.groupBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, List<V>> {\n  return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n }\n\n/**\n * Groups elements
of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline
fun <T, K, M : MutableMap<in K, MutableList<T>>> Array<out T>.groupByTo(destination: M, keySelector: (T) -
> K): M {\n  for (element in this) {\n    val key = keySelector(element)\n    val list = destination.getOrPut(key)
{ ArrayList<T>() }\n    list.add(element)\n  }\n  return destination\n }\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline
fun <K, M : MutableMap<in K, MutableList<Byte>>> ByteArray.groupByTo(destination: M, keySelector: (Byte) -
> K): M {\n  for (element in this) {\n    val key = keySelector(element)\n    val list = destination.getOrPut(key)
{ ArrayList<Byte>() }\n    list.add(element)\n  }\n  return destination\n }\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline
fun <K, M : MutableMap<in K, MutableList<Short>>> ShortArray.groupByTo(destination: M, keySelector: (Short)
-> K): M {\n  for (element in this) {\n    val key = keySelector(element)\n    val list =
destination.getOrPut(key) { ArrayList<Short>() }\n    list.add(element)\n  }\n  return destination\n }\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n *

```

```

@return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*/\npublic inline fun <K, M : MutableMap<in K, MutableList<Int>>> IntArray.groupByTo(destination: M,\n
keySelector: (Int) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list =\n
destination.getOrPut(key) { ArrayList<Int>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n *
Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each
element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n *
@return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*/\npublic inline fun <K, M : MutableMap<in K, MutableList<Long>>> LongArray.groupByTo(destination: M,\n
keySelector: (Long) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list =\n
destination.getOrPut(key) { ArrayList<Long>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n *
Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each
element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n *
@return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*/\npublic inline fun <K, M : MutableMap<in K, MutableList<Float>>> FloatArray.groupByTo(destination: M,\n
keySelector: (Float) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list =\n
destination.getOrPut(key) { ArrayList<Float>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n *
Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each
element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n *
@return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*/\npublic inline fun <K, M : MutableMap<in K, MutableList<Double>>> DoubleArray.groupByTo(destination: M,\n
keySelector: (Double) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list =\n
destination.getOrPut(key) { ArrayList<Double>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n *
Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each
element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n *
@return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*/\npublic inline fun <K, M : MutableMap<in K, MutableList<Boolean>>> BooleanArray.groupByTo(destination:\n
M, keySelector: (Boolean) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list\n
= destination.getOrPut(key) { ArrayList<Boolean>() }\n list.add(element)\n }\n return\n
destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector]\n
function\n * applied to each element and puts to the [destination] map each group key associated with a list of\n
corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample\n
samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M : MutableMap<in K,\n
MutableList<Char>>> CharArray.groupByTo(destination: M, keySelector: (Char) -> K): M {\n for (element in\n
this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<Char>() }\n\n
list.add(element)\n }\n return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function\n
applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to\n
the element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n
*\n * \n * @return The [destination] map.\n * \n * @sample\n
samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <T, K, V, M :  

MutableMap<in K, MutableList<V>>> Array<out T>.groupByTo(destination: M, keySelector: (T) -> K,\n
valueTransform: (T) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list =\n
destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return\n
destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the\n
original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the\n
[destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination]\n
map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic  

inline fun <K, V, M : MutableMap<in K, MutableList<V>>> ByteArray.groupByTo(destination: M, keySelector:\n
(Byte) -> K, valueTransform: (Byte) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n

```



```

    val list = destination.getOrPut(key) { ArrayList<V>() }
    list.add(valueTransform(element))
}
return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the
 * original array
 * by the key returned by the given [keySelector] function applied to the element
 * and puts to the [destination] map each group key associated with a list of corresponding values.
 *
 * @return The [destination] map.
 *
 * @sample samples.collections.Collections.Transformations.groupByKeyAndValues
 */
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> ShortArray.groupByTo(destination: M, keySelector: (Short) -> K, valueTransform: (Short) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the
 * original array
 * by the key returned by the given [keySelector] function applied to the element
 * and puts to the [destination] map each group key associated with a list of corresponding values.
 *
 * @return The [destination] map.
 *
 * @sample samples.collections.Collections.Transformations.groupByKeyAndValues
 */
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> IntArray.groupByTo(destination: M, keySelector: (Int) -> K, valueTransform: (Int) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the
 * original array
 * by the key returned by the given [keySelector] function applied to the element
 * and puts to the [destination] map each group key associated with a list of corresponding values.
 *
 * @return The [destination] map.
 *
 * @sample samples.collections.Collections.Transformations.groupByKeyAndValues
 */
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> LongArray.groupByTo(destination: M, keySelector: (Long) -> K, valueTransform: (Long) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the
 * original array
 * by the key returned by the given [keySelector] function applied to the element
 * and puts to the [destination] map each group key associated with a list of corresponding values.
 *
 * @return The [destination] map.
 *
 * @sample samples.collections.Collections.Transformations.groupByKeyAndValues
 */
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> FloatArray.groupByTo(destination: M, keySelector: (Float) -> K, valueTransform: (Float) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the
 * original array
 * by the key returned by the given [keySelector] function applied to the element
 * and puts to the [destination] map each group key associated with a list of corresponding values.
 *
 * @return The [destination] map.
 *
 * @sample samples.collections.Collections.Transformations.groupByKeyAndValues
 */
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> DoubleArray.groupByTo(destination: M, keySelector: (Double) -> K, valueTransform: (Double) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array
 * by the key returned by the given [keySelector] function applied to the element
 * and puts to the [destination] map each group key associated with a list of corresponding values.
 *
 * @return The [destination] map.
 *
 * @sample samples.collections.Collections.Transformations.groupByKeyAndValues
 */
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> BooleanArray.groupByTo(destination: M, keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the
 * original array
 * by the key returned by the given [keySelector] function applied to the element
 * and puts to the [destination] map each group key associated with a list of corresponding values.
 *
 * @return The [destination] map.
 *
 * @sample samples.collections.Collections.Transformations.groupByKeyAndValues
 */
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> BooleanArray.groupByTo(destination: M, keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the
 * original array
 * by the key returned by the given [keySelector] function applied to the element
 * and puts to the [destination] map each group key associated with a list of corresponding values.
 *
 * @return The [destination] map.
 *
 * @sample samples.collections.Collections.Transformations.groupByKeyAndValues
 */
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> BooleanArray.groupByTo(destination: M, keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

```

```

inline fun <K, V, M : MutableMap<in K, MutableList<V>>> CharArray.groupByTo(destination: M, keySelector:
(Char) -> K, valueTransform: (Char) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}
/**
 * Creates a [Grouping] source from an array to be used later with one of group-and-fold
operations
 * using the specified [keySelector] function to extract a key from each element.
 * @sample
samples.collections.Grouping.groupingByEachCount
 * @since Kotlin("1.1")
public inline fun <T, K>
Array<out T>.groupingBy(crossinline keySelector: (T) -> K): Grouping<T, K> {
    return object : Grouping<T,
K> {
        override fun sourceIterator(): Iterator<T> = this@groupingBy.iterator()
        override fun
keyOf(element: T): K = keySelector(element)
    }
}
/**
 * Returns a list containing the results of applying
the given [transform] function
 * to each element in the original array.
 * @sample
samples.collections.Collections.Transformations.map
 * @public inline fun <T, R> Array<out T>.map(transform:
(T) -> R): List<R> {
    return mapTo(ArrayList<R>(size), transform)
}
/**
 * Returns a list containing the
results of applying the given [transform] function
 * to each element in the original array.
 * @sample
samples.collections.Collections.Transformations.map
 * @public inline fun <R> ByteArray.map(transform: (Byte)
-> R): List<R> {
    return mapTo(ArrayList<R>(size), transform)
}
/**
 * Returns a list containing the
results of applying the given [transform] function
 * to each element in the original array.
 * @sample
samples.collections.Collections.Transformations.map
 * @public inline fun <R> ShortArray.map(transform:
(Short) -> R): List<R> {
    return mapTo(ArrayList<R>(size), transform)
}
/**
 * Returns a list containing
the results of applying the given [transform] function
 * to each element in the original array.
 * @sample
samples.collections.Collections.Transformations.map
 * @public inline fun <R> IntArray.map(transform: (Int) ->
R): List<R> {
    return mapTo(ArrayList<R>(size), transform)
}
/**
 * Returns a list containing the results
of applying the given [transform] function
 * to each element in the original array.
 * @sample
samples.collections.Collections.Transformations.map
 * @public inline fun <R> LongArray.map(transform:
(Long) -> R): List<R> {
    return mapTo(ArrayList<R>(size), transform)
}
/**
 * Returns a list containing
the results of applying the given [transform] function
 * to each element in the original array.
 * @sample
samples.collections.Collections.Transformations.map
 * @public inline fun <R> FloatArray.map(transform: (Float)
-> R): List<R> {
    return mapTo(ArrayList<R>(size), transform)
}
/**
 * Returns a list containing the
results of applying the given [transform] function
 * to each element in the original array.
 * @sample
samples.collections.Collections.Transformations.map
 * @public inline fun <R> DoubleArray.map(transform:
(Double) -> R): List<R> {
    return mapTo(ArrayList<R>(size), transform)
}
/**
 * Returns a list containing
the results of applying the given [transform] function
 * to each element in the original array.
 * @sample
samples.collections.Collections.Transformations.map
 * @public inline fun <R> BooleanArray.map(transform:
(Boolean) -> R): List<R> {
    return mapTo(ArrayList<R>(size), transform)
}
/**
 * Returns a list
containing the results of applying the given [transform] function
 * to each element in the original array.
 * @sample
samples.collections.Collections.Transformations.map
 * @public inline fun <R>
CharArray.map(transform: (Char) -> R): List<R> {
    return mapTo(ArrayList<R>(size), transform)
}
/**
 * Returns a list containing the results of applying the given [transform] function
 * to each element and its index in
the original array.
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 * @public inline fun <T, R> Array<out
T>.mapIndexed(transform: (index: Int, T) -> R): List<R> {
    return mapIndexedTo(ArrayList<R>(size),
transform)
}
/**
 * Returns a list containing the results of applying the given [transform] function
 * to each
element and its index in the original array.
 * @param [transform] function that takes the index of an element and
the element itself
 * and returns the result of the transform applied to the element.
 * @public inline fun <R>
ByteArray.mapIndexed(transform: (index: Int, Byte) -> R): List<R> {
    return
mapIndexedTo(ArrayList<R>(size), transform)
}
/**
 * Returns a list containing the results of applying the
given [transform] function
 * to each element and its index in the original array.
 * @param [transform] function
that takes the index of an element and the element itself
 * and returns the result of the transform applied to the
element.
 * @public inline fun <R> ShortArray.mapIndexed(transform: (index: Int, Short) -> R): List<R> {
}

```

```

return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying
the given [transform] function\n * to each element and its index in the original array.\n * @param [transform]
function that takes the index of an element and the element itself\n * and returns the result of the transform applied
to the element.\n */\npublic inline fun <R> IntArray.mapIndexed(transform: (index: Int, Int) -> R): List<R> {\n
return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying
the given [transform] function\n * to each element and its index in the original array.\n * @param [transform]
function that takes the index of an element and the element itself\n * and returns the result of the transform applied
to the element.\n */\npublic inline fun <R> LongArray.mapIndexed(transform: (index: Int, Long) -> R): List<R> {\n
return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying
the given [transform] function\n * to each element and its index in the original array.\n * @param [transform]
function that takes the index of an element and the element itself\n * and returns the result of the
transform applied to the element.\n */\npublic inline fun <R> FloatArray.mapIndexed(transform: (index: Int, Float) -
> R): List<R> {\n
return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing
the results of applying the given [transform] function\n * to each element and its index in the original array.\n *
@param [transform] function that takes the index of an element and the element itself\n * and returns the result of
the transform applied to the element.\n */\npublic inline fun <R> DoubleArray.mapIndexed(transform: (index: Int,
Double) -> R): List<R> {\n
return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list
containing the results of applying the given [transform] function\n * to each element and its index in the original
array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the
result of the transform applied to the element.\n */\npublic inline fun <R> BooleanArray.mapIndexed(transform:
(index: Int, Boolean) -> R): List<R> {\n
return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing
the results of applying the given [transform] function\n * to each element and its index in the original array.\n *
@param [transform] function that takes the index of an element and the element itself\n * and returns the result of
the transform applied to the element.\n */\npublic inline fun <R> CharArray.mapIndexed(transform: (index: Int, Char) -> R): List<R> {\n
return
mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing only the non-null results of
applying the given [transform] function\n * to each element and its index in the original array.\n * @param
[transform] function that takes the index of an element and the element itself\n * and returns the result of the
transform applied to the element.\n */\npublic inline fun <T, R : Any> Array<out
T>.mapIndexedNotNull(transform: (index: Int, T) -> R?): List<R> {\n
return
mapIndexedNotNullTo(ArrayList<R>(), transform)\n}\n\n/**\n * Applies the given [transform] function to each
element and its index in the original array\n * and appends only the non-null results to the given [destination].\n *
@param [transform] function that takes the index of an element and the element itself\n * and returns the result of
the transform applied to the element.\n */\npublic inline fun <T, R : Any, C : MutableCollection<in R>> Array<out
T>.mapIndexedNotNullTo(destination: C, transform: (index: Int, T) -> R?): C {\n
forEachIndexed { index,
element -> transform(index, element)?.let { destination.add(it) } }\n
return destination\n}\n\n/**\n * Applies the
given [transform] function to each element and its index in the original array\n * and appends the results to the given
[destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and
returns the result of the transform applied to the element.\n */\npublic inline fun <T, R, C : MutableCollection<in
R>> Array<out T>.mapIndexedTo(destination: C, transform: (index: Int, T) -> R): C {\n
var index = 0\n
for (item in this)\n
destination.add(transform(index++, item))\n
return destination\n}\n\n/**\n * Applies the given
[transform] function to each element and its index in the original array\n * and appends the results to the given
[destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and
returns the result of the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in R>>
ByteArray.mapIndexedTo(destination: C, transform: (index: Int, Byte) -> R): C {\n
var index = 0\n
for (item in
this)\n
destination.add(transform(index++, item))\n
return destination\n}\n\n/**\n * Applies the given
[transform] function to each element and its index in the original array\n * and appends the results to the given
[destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and

```

returns the result of the transform applied to the element.

```

public inline fun <R, C : MutableCollection<in R>>
ShortArray.mapIndexedTo(destination: C, transform: (index: Int, Short) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

```

* Applies the given [transform] function to each element and its index in the original array * and appends the results to the given [destination].

* @param [transform] function that takes the index of an element and the element itself * and returns the result of the transform applied to the element.

```

public inline fun <R, C : MutableCollection<in R>>
IntArray.mapIndexedTo(destination: C, transform: (index: Int, Int) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

```

* Applies the given [transform] function to each element and its index in the original array * and appends the results to the given [destination].

* @param [transform] function that takes the index of an element and the element itself * and returns the result of the transform applied to the element.

```

public inline fun <R, C : MutableCollection<in R>>
LongArray.mapIndexedTo(destination: C, transform: (index: Int, Long) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

```

* Applies the given [transform] function to each element and its index in the original array * and appends the results to the given [destination].

* @param [transform] function that takes the index of an element and the element itself * and returns the result of the transform applied to the element.

```

public inline fun <R, C : MutableCollection<in R>>
FloatArray.mapIndexedTo(destination: C, transform: (index: Int, Float) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

```

* Applies the given [transform] function to each element and its index in the original array * and appends the results to the given [destination].

* @param [transform] function that takes the index of an element and the element itself * and returns the result of the transform applied to the element.

```

public inline fun <R, C : MutableCollection<in R>>
DoubleArray.mapIndexedTo(destination: C, transform: (index: Int, Double) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

```

* Applies the given [transform] function to each element and its index in the original array * and appends the results to the given [destination].

* @param [transform] function that takes the index of an element and the element itself * and returns the result of the transform applied to the element.

```

public inline fun <R, C : MutableCollection<in R>>
BooleanArray.mapIndexedTo(destination: C, transform: (index: Int, Boolean) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

```

* Applies the given [transform] function to each element and its index in the original array * and appends the results to the given [destination].

* @param [transform] function that takes the index of an element and the element itself * and returns the result of the transform applied to the element.

```

public inline fun <R, C : MutableCollection<in R>>
CharArray.mapIndexedTo(destination: C, transform: (index: Int, Char) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

```

* Returns a list containing only the non-null results of applying the given [transform] function * to each element in the original array.

@sample samples.collections.Collections.Transformations.mapNotNull

```

public inline fun <T, R : Any>
Array<out T>.mapNotNull(transform: (T) -> R?): List<R> {
    return mapNotNullTo(ArrayList<R>(),
        transform)
}

```

* Applies the given [transform] function to each element in the original array * and appends only the non-null results to the given [destination].

```

public inline fun <T, R : Any, C :
MutableCollection<in R>>
Array<out T>.mapNotNullTo(destination: C, transform: (T) -> R?): C {
    for (element in this)
        let { destination.add(transform(element)) }
    return destination
}

```

* Applies the given [transform] function to each element of the original array * and appends the results to the given [destination].

```

public inline fun <T, R, C : MutableCollection<in R>>
Array<out T>.mapTo(destination: C, transform: (T) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

```

* Applies the given [transform] function to each element of the original array * and appends the results to the given [destination].

```

public inline fun <R, C : MutableCollection<in R>>
ByteArray.mapTo(destination: C,
transform: (Byte) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

```

* Applies the given [transform] function to each element of the original array * and appends the results to the given [destination].

```

public inline fun <R, C : MutableCollection<in R>>

```

`ShortArray.mapTo(destination: C, transform: (Short) -> R): C` {
 for (item in this)
 destination.add(transform(item))
 return destination
}

Applies the given [transform] function to each element of the original array and appends the results to the given [destination].

`public inline fun <R, C : MutableCollection<in R>> IntArray.mapTo(destination: C, transform: (Int) -> R): C` {
 for (item in this)
 destination.add(transform(item))
 return destination
}

Applies the given [transform] function to each element of the original array and appends the results to the given [destination].

`public inline fun <R, C : MutableCollection<in R>> LongArray.mapTo(destination: C, transform: (Long) -> R): C` {
 for (item in this)
 destination.add(transform(item))
 return destination
}

Applies the given [transform] function to each element of the original array and appends the results to the given [destination].

`public inline fun <R, C : MutableCollection<in R>> FloatArray.mapTo(destination: C, transform: (Float) -> R): C` {
 for (item in this)
 destination.add(transform(item))
 return destination
}

Applies the given [transform] function to each element of the original array and appends the results to the given [destination].

`public inline fun <R, C : MutableCollection<in R>> DoubleArray.mapTo(destination: C, transform: (Double) -> R): C` {
 for (item in this)
 destination.add(transform(item))
 return destination
}

Applies the given [transform] function to each element of the original array and appends the results to the given [destination].

`public inline fun <R, C : MutableCollection<in R>> BooleanArray.mapTo(destination: C, transform: (Boolean) -> R): C` {
 for (item in this)
 destination.add(transform(item))
 return destination
}

Applies the given [transform] function to each element of the original array and appends the results to the given [destination].

`public inline fun <R, C : MutableCollection<in R>> CharArray.mapTo(destination: C, transform: (Char) -> R): C` {
 for (item in this)
 destination.add(transform(item))
 return destination
}

Returns a lazy [Iterable] that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

`public fun <T> Array<out T>.withIndex(): Iterable<IndexedValue<T>>` {
 return IndexingIterable { iterator() }
}

Returns a lazy [Iterable] that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

`public fun ByteArray.withIndex(): Iterable<IndexedValue<Byte>>` {
 return IndexingIterable { iterator() }
}

Returns a lazy [Iterable] that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

`public fun ShortArray.withIndex(): Iterable<IndexedValue<Short>>` {
 return IndexingIterable { iterator() }
}

Returns a lazy [Iterable] that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

`public fun IntArray.withIndex(): Iterable<IndexedValue<Int>>` {
 return IndexingIterable { iterator() }
}

Returns a lazy [Iterable] that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

`public fun LongArray.withIndex(): Iterable<IndexedValue<Long>>` {
 return IndexingIterable { iterator() }
}

Returns a lazy [Iterable] that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

`public fun FloatArray.withIndex(): Iterable<IndexedValue<Float>>` {
 return IndexingIterable { iterator() }
}

Returns a lazy [Iterable] that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

`public fun DoubleArray.withIndex(): Iterable<IndexedValue<Double>>` {
 return IndexingIterable { iterator() }
}

Returns a lazy [Iterable] that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

`public fun BooleanArray.withIndex(): Iterable<IndexedValue<Boolean>>` {
 return IndexingIterable { iterator() }
}

Returns a lazy [Iterable] that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

`public fun CharArray.withIndex(): Iterable<IndexedValue<Char>>` {
 return IndexingIterable { iterator() }
}

Returns a list containing only distinct elements from the given array.

Among equal elements of the given array, only the first one will be present in the resulting list.

The elements in the resulting list are in the same order as they were in the source array.

`@sample samples.collections.Collections.Transformations.distinctAndDistinctBy`

`public fun <T> Array<out T>.distinct(): List<T>` {
 return this.toMutableSet().toList()
}

Returns a list

containing only distinct elements from the given array.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample

```

samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^\npublic fun ByteArray.distinct():
List<Byte> {\n    return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only distinct elements
from the given array.\n * \n * The elements in the resulting list are in the same order as they were in the source
array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^\npublic fun
ShortArray.distinct(): List<Short> {\n    return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing
only distinct elements from the given array.\n * \n * The elements in the resulting list are in the same order as they
were in the source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^\npublic fun IntArray.distinct():
List<Int> {\n    return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only distinct elements from
the given array.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n *
\n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^\npublic fun
LongArray.distinct(): List<Long> {\n    return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing
only distinct elements from the given array.\n * \n * The elements in the resulting list are in the same order as they
were in the source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^\npublic fun FloatArray.distinct():
List<Float> {\n    return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only distinct elements
from the given array.\n * \n * The elements in the resulting list are in the same order as they were in the source
array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^\npublic fun
DoubleArray.distinct(): List<Double> {\n    return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list
containing only distinct elements from the given array.\n * \n * The elements in the resulting list are in the same
order as they were in the source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^\npublic fun BooleanArray.distinct():
List<Boolean> {\n    return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only distinct
elements from the given array.\n * \n * The elements in the resulting list are in the same order as they were in the
source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^\npublic inline fun <T, K> Array<out
T>.distinctBy(selector: (T) -> K): List<T> {\n    val set = HashSet<K>()\n    val list = ArrayList<T>()\n    for (e in
this) {\n        val key = selector(e)\n        if (set.add(key))\n            list.add(e)\n    }\n    return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys returned by the given [selector] function.\n * \n * Among elements of the given array with equal keys, only the first one will be present in the resulting list.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^\npublic inline fun <T, K> Array<out
T>.distinctBy(selector: (T) -> K): List<T> {\n    val set = HashSet<K>()\n    val list = ArrayList<T>()\n    for (e in
this) {\n        val key = selector(e)\n        if (set.add(key))\n            list.add(e)\n    }\n    return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^\npublic inline fun <K>
ByteArray.distinctBy(selector: (Byte) -> K): List<Byte> {\n    val set = HashSet<K>()\n    val list =
ArrayList<Byte>()\n    for (e in this) {\n        val key = selector(e)\n        if (set.add(key))\n            list.add(e)\n    }\n    return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^\npublic inline fun <K>
ShortArray.distinctBy(selector: (Short) -> K): List<Short> {\n    val set = HashSet<K>()\n    val list =
ArrayList<Short>()\n    for (e in this) {\n        val key = selector(e)\n        if (set.add(key))\n            list.add(e)\n    }\n    return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample

```

```

samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^\npublic inline fun <K>
IntArray.distinctBy(selector: (Int) -> K): List<Int> {\n    val set = HashSet<K>()\n    val list = ArrayList<Int>()\n
for (e in this) {\n    val key = selector(e)\n    if (set.add(key))\n        list.add(e)\n    }\n    return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys returned by the given
[selector] function.\n * \n * The elements in the resulting list are in the same order as they were in the source
array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^\npublic inline
fun <K> LongArray.distinctBy(selector: (Long) -> K): List<Long> {\n    val set = HashSet<K>()\n    val list =
ArrayList<Long>()\n    for (e in this) {\n    val key = selector(e)\n    if (set.add(key))\n        list.add(e)\n    }\n
return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys
returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they
were in the source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^\npublic inline fun <K>
FloatArray.distinctBy(selector: (Float) -> K): List<Float> {\n    val set = HashSet<K>()\n    val list =
ArrayList<Float>()\n    for (e in this) {\n    val key = selector(e)\n    if (set.add(key))\n        list.add(e)\n    }\n
return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys
returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they
were in the source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^\npublic inline fun <K>
DoubleArray.distinctBy(selector: (Double) -> K): List<Double> {\n    val set = HashSet<K>()\n    val list =
ArrayList<Double>()\n    for (e in this) {\n    val key = selector(e)\n    if (set.add(key))\n        list.add(e)\n
    }\n    return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys
returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they
were in the source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^\npublic inline fun <K>
BooleanArray.distinctBy(selector: (Boolean) -> K): List<Boolean> {\n    val set = HashSet<K>()\n    val list =
ArrayList<Boolean>()\n    for (e in this) {\n    val key = selector(e)\n    if (set.add(key))\n        list.add(e)\n
    }\n    return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys
returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they
were in the source array.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^\npublic inline fun <K>
CharArray.distinctBy(selector: (Char) -> K): List<Char> {\n    val set = HashSet<K>()\n    val list =
ArrayList<Char>()\n    for (e in this) {\n    val key = selector(e)\n    if (set.add(key))\n        list.add(e)\n    }\n
return list\n}\n\n/**\n * Returns a set containing all elements that are contained by both this array and the specified
collection.\n * \n * The returned set preserves the element iteration order of the original array.\n * \n * To get a set
containing all elements that are contained at least in one of these collections use [union].\n *^\npublic infix fun <T>
Array<out T>.intersect(other: Iterable<T>): Set<T> {\n    val set = this.toMutableSet()\n    set.retainAll(other)\n
return set\n}\n\n/**\n * Returns a set containing all elements that are contained by both this array and the specified
collection.\n * \n * The returned set preserves the element iteration order of the original array.\n * \n * To get a set
containing all elements that are contained at least in one of these collections use [union].\n *^\npublic infix fun
ByteArray.intersect(other: Iterable<Byte>): Set<Byte> {\n    val set = this.toMutableSet()\n    set.retainAll(other)\n
return set\n}\n\n/**\n * Returns a set containing all elements that are contained by both this array and the specified
collection.\n * \n * The returned set preserves the element iteration order of the original array.\n * \n * To get a set
containing all elements that are contained at least in one of these collections use [union].\n *^\npublic infix fun
ShortArray.intersect(other: Iterable<Short>): Set<Short> {\n    val set = this.toMutableSet()\n    set.retainAll(other)\n
return set\n}\n\n/**\n * Returns a set containing all elements that are contained by both this array and the specified
collection.\n * \n * The returned set preserves the element iteration order of the original array.\n * \n * To get a set
containing all elements that are contained at least in one of these collections use
[union].\n *^\npublic infix fun IntArray.intersect(other: Iterable<Int>): Set<Int> {\n    val set = this.toMutableSet()\n

```

```

    set.retainAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that are contained by both this
array and the specified collection.\n * \n * The returned set preserves the element iteration order of the original
array.\n * \n * To get a set containing all elements that are contained at least in one of these collections use
[union].\n */\npublic infix fun LongArray.intersect(other: Iterable<Long>): Set<Long> {\n    val set =
this.toMutableSet()\n    set.retainAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that are
contained by both this array and the specified collection.\n * \n * The returned set preserves the element iteration
order of the original array.\n * \n * To get a set containing all elements that are contained at least in one of these
collections use [union].\n */\npublic infix fun FloatArray.intersect(other: Iterable<Float>): Set<Float> {\n    val set =
this.toMutableSet()\n    set.retainAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that
are contained by both this array and the specified collection.\n * \n * The returned set preserves the element iteration
order of the original array.\n * \n * To get a set containing all elements that are contained at least in one of these
collections use [union].\n */\npublic infix fun DoubleArray.intersect(other: Iterable<Double>): Set<Double> {\n    val set =
this.toMutableSet()\n    set.retainAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all
elements that are contained by both this array and the specified collection.\n * \n * The returned set preserves the
element iteration order of the original array.\n * \n * To get a set containing all elements that are contained at least
in one of these collections use [union].\n */\npublic infix fun BooleanArray.intersect(other: Iterable<Boolean>):
Set<Boolean> {\n    val set = this.toMutableSet()\n    set.retainAll(other)\n    return set\n}\n\n/**\n * Returns a set
containing all elements that are contained by both this array and the specified collection.\n * \n * The returned set
preserves the element iteration order of the original array.\n * \n * To get a set containing all elements that are
contained at least in one of these collections use [union].\n */\npublic infix fun CharArray.intersect(other:
Iterable<Char>): Set<Char> {\n    val set = this.toMutableSet()\n    set.retainAll(other)\n    return set\n}\n\n/**\n *
Returns a set containing all elements that are contained by this array and not contained by the specified collection.\n *
\n * The returned set preserves the element iteration order of the original array.\n */\npublic infix fun <T>
Array<out T>.subtract(other: Iterable<T>): Set<T> {\n    val set = this.toMutableSet()\n    set.removeAll(other)\n
return set\n}\n\n/**\n * Returns a set containing all elements that are contained by this array and not contained by
the specified collection.\n * \n * The returned set preserves the element iteration order of the original array.\n */\n
public infix fun ByteArray.subtract(other: Iterable<Byte>): Set<Byte> {\n    val set = this.toMutableSet()\n    set.removeAll(other)\n
return set\n}\n\n/**\n * Returns a set containing all elements that are contained by this array and not contained by
the specified collection.\n * \n * The returned set preserves the element iteration order of the original array.\n */\n
public infix fun ShortArray.subtract(other: Iterable<Short>): Set<Short> {\n    val set =
this.toMutableSet()\n    set.removeAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that
are contained by this array and not contained by the specified collection.\n * \n * The returned set preserves the
element iteration order of the original array.\n */\npublic infix fun IntArray.subtract(other: Iterable<Int>): Set<Int>
{\n    val set = this.toMutableSet()\n    set.removeAll(other)\n    return set\n}\n\n/**\n * Returns a set containing
all elements that are contained by this array and not contained by the specified collection.\n * \n * The returned set
preserves the element iteration order of the original array.\n */\npublic infix fun LongArray.subtract(other:
Iterable<Long>): Set<Long> {\n    val set = this.toMutableSet()\n    set.removeAll(other)\n    return set\n}\n\n/**\n *
Returns a set containing all elements that are contained by this array and not contained by the specified
collection.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic infix fun
FloatArray.subtract(other: Iterable<Float>): Set<Float> {\n    val set = this.toMutableSet()\n    set.removeAll(other)\n
return set\n}\n\n/**\n * Returns a set containing all elements that are contained by this array and not contained by
the specified collection.\n * \n * The returned set preserves the element iteration order of the original array.\n */\n
public infix fun DoubleArray.subtract(other: Iterable<Double>): Set<Double> {\n    val set =
this.toMutableSet()\n    set.removeAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that
are contained by this array and not contained by the specified collection.\n * \n * The returned set preserves the
element iteration order of the original array.\n */\npublic infix fun BooleanArray.subtract(other: Iterable<Boolean>):
Set<Boolean> {\n    val set = this.toMutableSet()\n    set.removeAll(other)\n    return set\n}\n\n/**\n * Returns a set
containing all elements that are contained by this array and not contained by the specified collection.\n * \n * The

```


returned set preserves the element iteration order of the original array.

```

public infix fun
CharArray.subtract(other: Iterable<Char>): Set<Char> {
    val set = this.toMutableSet()
    set.removeAll(other)
    return set
}

```

* Returns a new [MutableSet] containing all distinct elements from the given array. * The returned set preserves the element iteration order of the original array.

```

public fun <T> Array<out
T>.toMutableSet(): MutableSet<T> {
    return toCollection(LinkedHashSet<T>(mapCapacity(size)))
}

```

* Returns a new [MutableSet] containing all distinct elements from the given array. * The returned set preserves the element iteration order of the original array.

```

public fun ByteArray.toMutableSet():
MutableSet<Byte> {
    return toCollection(LinkedHashSet<Byte>(mapCapacity(size)))
}

```

* Returns a new [MutableSet] containing all distinct elements from the given array. * The returned set preserves the element iteration order of the original array.

```

public fun ShortArray.toMutableSet(): MutableSet<Short> {
    return toCollection(LinkedHashSet<Short>(mapCapacity(size)))
}

```

* Returns a new [MutableSet] containing all distinct elements from the given array. * The returned set preserves the element iteration order of the original array.

```

public fun IntArray.toMutableSet(): MutableSet<Int> {
    return toCollection(LinkedHashSet<Int>(mapCapacity(size)))
}

```

* Returns a new [MutableSet] containing all distinct elements from the given array. * The returned set preserves the element iteration order of the original array.

```

public fun LongArray.toMutableSet(): MutableSet<Long> {
    return toCollection(LinkedHashSet<Long>(mapCapacity(size)))
}

```

* Returns a new [MutableSet] containing all distinct elements from the given array. * The returned set preserves the element iteration order of the original array.

```

public fun FloatArray.toMutableSet(): MutableSet<Float> {
    return toCollection(LinkedHashSet<Float>(mapCapacity(size)))
}

```

* Returns a new [MutableSet] containing all distinct elements from the given array. * The returned set preserves the element iteration order of the original array.

```

public fun DoubleArray.toMutableSet(): MutableSet<Double> {
    return toCollection(LinkedHashSet<Double>(mapCapacity(size)))
}

```

* Returns a new [MutableSet] containing all distinct elements from the given array. * The returned set preserves the element iteration order of the original array.

```

public fun BooleanArray.toMutableSet(): MutableSet<Boolean> {
    return toCollection(LinkedHashSet<Boolean>(mapCapacity(size)))
}

```

* Returns a new [MutableSet] containing all distinct elements from the given array. * The returned set preserves the element iteration order of the original array.

```

public fun CharArray.toMutableSet(): MutableSet<Char> {
    return toCollection(LinkedHashSet<Char>(mapCapacity(size.coerceAtMost(128))))
}

```

* Returns a set containing all distinct elements from both collections. * The returned set preserves the element iteration order of the original array. * Those elements of the [other] collection that are unique are iterated in the end * in the order of the [other] collection. * To get a set containing all elements that are contained in both collections use [intersect].

```

public infix fun <T> Array<out T>.union(other: Iterable<T>): Set<T> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

```

* Returns a set containing all distinct elements from both collections. * The returned set preserves the element iteration order of the original array. * Those elements of the [other] collection that are unique are iterated in the end * in the order of the [other] collection. * To get a set containing all elements that are contained in both collections use [intersect].

```

public infix fun ByteArray.union(other: Iterable<Byte>): Set<Byte> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

```

* Returns a set containing all distinct elements from both collections. * The returned set preserves the element iteration order of the original array. * Those elements of the [other] collection that are unique are iterated in the end * in the order of the [other] collection. * To get a set containing all elements that are contained in both collections use [intersect].

```

public infix fun ShortArray.union(other: Iterable<Short>): Set<Short> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

```

* Returns a set containing all distinct elements from both collections. * The returned set preserves the element iteration order of the original array. * Those elements of the [other] collection that are unique are iterated in the end * in the order of the [other] collection. * To get a set containing all elements that are contained in both collections use [intersect].

```

public infix fun IntArray.union(other: Iterable<Int>): Set<Int> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

```

* Returns a set containing all distinct elements

from both collections.

- * The returned set preserves the element iteration order of the original array.
- * Those elements of the [other] collection that are unique are iterated in the end
- * in the order of the [other] collection.

To get a set containing all elements that are contained in both collections use [intersect].

```

public infix fun LongArray.union(other: Iterable<Long>): Set<Long> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

```

Returns a set containing all distinct elements from both collections.

- * The returned set preserves the element iteration order of the original array.
- * Those elements of the [other] collection that are unique are iterated in the end
- * in the order of the [other] collection.

To get a set containing all elements that are contained in both collections use [intersect].

```

public infix fun FloatArray.union(other: Iterable<Float>): Set<Float> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

```

Returns a set containing all distinct elements from both collections.

- * The returned set preserves the element iteration order of the original array.
- * Those elements of the [other] collection that are unique are iterated in the end
- * in the order of the [other] collection.

To get a set containing all elements that are contained in both collections use [intersect].

```

public infix fun DoubleArray.union(other: Iterable<Double>): Set<Double> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

```

Returns a set containing all distinct elements from both collections.

- * The returned set preserves the element iteration order of the original array.
- * Those elements of the [other] collection that are unique are iterated in the end
- * in the order of the [other] collection.

To get a set containing all elements that are contained in both collections use [intersect].

```

public infix fun BooleanArray.union(other: Iterable<Boolean>): Set<Boolean> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

```

Returns a set containing all distinct elements from both collections.

- * The returned set preserves the element iteration order of the original array.
- * Those elements of the [other] collection that are unique are iterated in the end
- * in the order of the [other] collection.

To get a set containing all elements that are contained in both collections use [intersect].

```

public infix fun CharArray.union(other: Iterable<Char>): Set<Char> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

```

Returns `true` if all elements match the given [predicate].

- * @sample samples.collections.Collections.Aggregates.all

```

public inline fun <T> Array<out T>.all(predicate: (T) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

```

Returns `true` if all elements match the given [predicate].

- * @sample samples.collections.Collections.Aggregates.all

```

public inline fun ByteArray.all(predicate: (Byte) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

```

Returns `true` if all elements match the given [predicate].

- * @sample samples.collections.Collections.Aggregates.all

```

public inline fun ShortArray.all(predicate: (Short) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

```

Returns `true` if all elements match the given [predicate].

- * @sample samples.collections.Collections.Aggregates.all

```

public inline fun IntArray.all(predicate: (Int) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

```

Returns `true` if all elements match the given [predicate].

- * @sample samples.collections.Collections.Aggregates.all

```

public inline fun LongArray.all(predicate: (Long) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

```

Returns `true` if all elements match the given [predicate].

- * @sample samples.collections.Collections.Aggregates.all

```

public inline fun FloatArray.all(predicate: (Float) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

```

Returns `true` if all elements match the given [predicate].

- * @sample samples.collections.Collections.Aggregates.all

```

public inline fun DoubleArray.all(predicate: (Double) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

```

Returns `true` if all elements match the given [predicate].

- * @sample samples.collections.Collections.Aggregates.all

```

public inline fun BooleanArray.all(predicate: (Boolean) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

```

Returns `true` if all elements match the given [predicate].

- * @sample samples.collections.Collections.Aggregates.all

```

public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

```

Returns `true` if

```

array has at least one element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n * \n\npublic fun
<T> Array<out T>.any(): Boolean {\n    return !isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one
element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n * \n\npublic fun ByteArray.any():
Boolean {\n    return !isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n * \n\npublic fun ShortArray.any(): Boolean {\n    return
!isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n * \n\npublic fun IntArray.any(): Boolean {\n    return
!isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n * \n\npublic fun LongArray.any(): Boolean {\n    return
!isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n * \n\npublic fun FloatArray.any(): Boolean {\n    return
!isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n * \n\npublic fun DoubleArray.any(): Boolean {\n    return
!isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n * \n\npublic fun BooleanArray.any(): Boolean {\n    return
!isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n * \n\npublic fun CharArray.any(): Boolean {\n    return
!isEmpty()\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n * \n\npublic inline fun <T> Array<out
T>.any(predicate: (T) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return true\n    return
false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n * \n\npublic inline fun ByteArray.any(predicate:
(Byte) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return true\n    return
false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n * \n\npublic inline fun ShortArray.any(predicate:
(Short) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return true\n    return
false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n * \n\npublic inline fun IntArray.any(predicate: (Int) -
> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return true\n    return false\n}\n\n/**\n *
Returns `true` if at least one element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n * \n\npublic inline fun LongArray.any(predicate:
(Long) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return true\n    return
false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n * \n\npublic inline fun FloatArray.any(predicate:
(Float) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return true\n    return
false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n * \n\npublic inline fun DoubleArray.any(predicate:
(Double) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return true\n    return
false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n * \n\npublic inline fun BooleanArray.any(predicate:
(Boolean) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return true\n    return
false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n * \n\npublic inline fun CharArray.any(predicate:
(Char) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return true\n    return
false\n}\n\n/**\n * Returns the number of elements in this array.\n * \n * \n * @kotlin.internal.InlineOnly\n * \n\npublic inline
fun <T> Array<out T>.count(): Int {\n    return size\n}\n\n/**\n * Returns the number of elements in this array.\n * \n *
\n * @kotlin.internal.InlineOnly\n * \n\npublic inline fun ByteArray.count(): Int {\n    return size\n}\n\n/**\n * Returns the
number of elements in this array.\n * \n * \n * @kotlin.internal.InlineOnly\n * \n\npublic inline fun ShortArray.count(): Int {\n

```

```

return size\n}\n\n/**\n * Returns the number of elements in this array.\n */\n@kotlin.internal.InlineOnly\npublic
inline fun IntArray.count(): Int {\n    return size\n}\n\n/**\n * Returns the number of elements in this array.\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.count(): Int {\n    return size\n}\n\n/**\n * Returns the
number of elements in this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.count(): Int {\n
return size\n}\n\n/**\n * Returns the number of elements in this array.\n */\n@kotlin.internal.InlineOnly\npublic
inline fun DoubleArray.count(): Int {\n    return size\n}\n\n/**\n * Returns the number of elements in this array.\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.count(): Int {\n    return size\n}\n\n/**\n * Returns
the number of elements in this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.count(): Int {\n
return size\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n */\npublic inline fun
<T> Array<out T>.count(predicate: (T) -> Boolean): Int {\n    var count = 0\n    for (element in this) if
(predicate(element)) ++count\n    return count\n}\n\n/**\n * Returns the number of elements matching the given
[predicate].\n */\npublic inline fun ByteArray.count(predicate: (Byte) -> Boolean): Int {\n    var count = 0\n    for
(element in this) if (predicate(element)) ++count\n    return count\n}\n\n/**\n * Returns the number of elements
matching the given [predicate].\n */\npublic inline fun ShortArray.count(predicate: (Short) -> Boolean): Int {\n    var
count = 0\n    for (element in this) if (predicate(element)) ++count\n    return count\n}\n\n/**\n * Returns the
number of elements matching the given [predicate].\n */\npublic inline fun IntArray.count(predicate: (Int) ->
Boolean): Int {\n    var count = 0\n    for (element in this) if (predicate(element)) ++count\n    return
count\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n */\npublic inline fun
LongArray.count(predicate: (Long) -> Boolean): Int {\n    var count = 0\n    for (element in this) if
(predicate(element)) ++count\n    return count\n}\n\n/**\n * Returns the number of elements matching the given
[predicate].\n */\npublic inline fun FloatArray.count(predicate: (Float) -> Boolean): Int {\n    var count = 0\n    for
(element in this) if (predicate(element)) ++count\n    return count\n}\n\n/**\n * Returns the number of elements
matching the given [predicate].\n */\npublic inline fun DoubleArray.count(predicate: (Double) -> Boolean): Int {\n
var count = 0\n    for (element in this) if (predicate(element)) ++count\n    return count\n}\n\n/**\n * Returns the
number of elements matching the given [predicate].\n */\npublic inline fun BooleanArray.count(predicate: (Boolean)
-> Boolean): Int {\n    var count = 0\n    for (element in this) if (predicate(element)) ++count\n    return
count\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n */\npublic inline fun
CharArray.count(predicate: (Char) -> Boolean): Int {\n    var count = 0\n    for (element in this) if
(predicate(element)) ++count\n    return count\n}\n\n/**\n * Accumulates value starting with [initial] value and
applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the
specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator
value and an element, and calculates the next accumulator value.\n */\npublic inline fun <T, R> Array<out
T>.fold(initial: R, operation: (acc: R, T) -> R): R {\n    var accumulator = initial\n    for (element in this)
accumulator = operation(accumulator, element)\n    return accumulator\n}\n\n/**\n * Accumulates value starting
with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n
*/\n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes
current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <R>
ByteArray.fold(initial: R, operation: (acc: R, Byte) -> R): R {\n    var accumulator = initial\n    for (element in
this) accumulator = operation(accumulator, element)\n    return accumulator\n}\n\n/**\n * Accumulates value starting
with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n
*/\n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes
current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <R>
ShortArray.fold(initial: R, operation: (acc: R, Short) -> R): R {\n    var accumulator = initial\n    for (element in
this) accumulator = operation(accumulator, element)\n    return accumulator\n}\n\n/**\n * Accumulates value starting
with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n
*/\n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes
current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <R>
IntArray.fold(initial: R, operation: (acc: R, Int) -> R): R {\n    var accumulator = initial\n    for (element in
this)

```

```

accumulator = operation(accumulator, element)\n  return accumulator\n}\n\n/**\n * Accumulates value starting
with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n *
\n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes
current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <R>
LongArray.fold(initial: R, operation: (acc: R, Long) -> R): R {\n  var accumulator = initial\n  for (element in this)
accumulator = operation(accumulator, element)\n  return accumulator\n}\n\n/**\n * Accumulates value starting
with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n *
\n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes
current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <R>
FloatArray.fold(initial: R, operation: (acc: R, Float) -> R): R {\n  var accumulator = initial\n  for (element in this)
accumulator = operation(accumulator, element)\n  return accumulator\n}\n\n/**\n * Accumulates value starting
with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n *
\n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes
current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <R>
DoubleArray.fold(initial: R, operation: (acc: R, Double) -> R): R {\n  var accumulator = initial\n  for (element in
this) accumulator = operation(accumulator, element)\n  return accumulator\n}\n\n/**\n * Accumulates value
starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each
element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that
takes current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun
<R> BooleanArray.fold(initial: R, operation: (acc: R, Boolean) -> R): R {\n  var accumulator = initial\n  for
(element in this) accumulator = operation(accumulator, element)\n  return accumulator\n}\n\n/**\n * Accumulates
value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and
each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation]
function that takes current accumulator value and an element, and calculates the next accumulator value.\n */\n
*\npublic inline fun <R> CharArray.fold(initial: R, operation: (acc: R, Char) -> R): R {\n  var accumulator =
initial\n  for (element in this) accumulator = operation(accumulator, element)\n  return accumulator\n}\n\n/**\n *
Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator
value and each element with its index in the original array.\n * \n * Returns the specified [initial] value if the array
is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and
the element itself, and calculates the next accumulator value.\n */\npublic inline fun <T, R> Array<out
T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {\n  var index = 0\n  var accumulator =
initial\n  for (element in this) accumulator = operation(index++, accumulator, element)\n  return
accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to
right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns the
specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an
element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n */\n
*\npublic inline fun <R> ByteArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Byte) -> R): R {\n  var
index = 0\n  var accumulator = initial\n  for (element in this) accumulator = operation(index++, accumulator,
element)\n  return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying
[operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n
*\n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the
index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator
value.\n */\npublic inline fun <R> ShortArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): R
{\n  var index = 0\n  var accumulator = initial\n  for (element in this) accumulator = operation(index++,
accumulator, element)\n  return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and
applying [operation] from left to right\n * to current accumulator value and each element with its index in the
original array.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the

```

```

next accumulator value.\n */\npublic inline fun <R> IntArray.foldIndexed(initial: R, operation: (index: Int, acc: R,
Int) -> R): R {\n  var index = 0\n  var accumulator = initial\n  for (element in this) accumulator =
operation(index++, accumulator, element)\n  return accumulator\n}\n\n/**\n * Accumulates value starting with
[initial] value and applying [operation] from left to right\n * to current accumulator value and each element with its
index in the original array.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param
[operation] function that takes the index of an element, current accumulator value\n * and the element itself, and
calculates the next accumulator value.\n */\npublic inline fun <R> LongArray.foldIndexed(initial: R, operation:
(index: Int, acc: R, Long) -> R): R {\n  var index = 0\n  var accumulator = initial\n  for (element in this)
accumulator = operation(index++, accumulator, element)\n  return accumulator\n}\n\n/**\n * Accumulates value
starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each
element with its index in the original array.\n * \n * Returns the specified [initial] value if the array is empty.\n *
\n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element
itself, and calculates the next accumulator value.\n */\npublic inline fun <R> FloatArray.foldIndexed(initial: R,
operation: (index: Int, acc: R, Float) -> R): R {\n  var index = 0\n  var accumulator = initial\n  for (element in
this) accumulator = operation(index++, accumulator, element)\n  return accumulator\n}\n\n/**\n * Accumulates
value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and
each element with its index in the original array.\n * \n * Returns the specified [initial] value if the array is empty.\n *
\n * @param [operation] function that takes the index of an element, current accumulator value\n * and the
element itself, and calculates the next accumulator value.\n */\npublic inline fun <R>
DoubleArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Double) -> R): R {\n  var index = 0\n  var
accumulator = initial\n  for (element in this) accumulator = operation(index++, accumulator, element)\n  return
accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to
right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns the
specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an
element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n */\npublic
inline fun <R> BooleanArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Boolean) -> R): R {\n  var
index = 0\n  var accumulator = initial\n  for (element in this) accumulator = operation(index++, accumulator,
element)\n  return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying
[operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n *
\n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the
index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator
value.\n */\npublic inline fun <R> CharArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): R
{\n  var index = 0\n  var accumulator = initial\n  for (element in this) accumulator = operation(index++,
accumulator, element)\n  return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and
applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the
specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current
accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <T, R> Array<out
T>.foldRight(initial: R, operation: (T, acc: R) -> R): R {\n  var index = lastIndex\n  var accumulator = initial\n
while (index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n  }\n  return
accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to
left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is
empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the
next accumulator value.\n */\npublic inline fun <R> ByteArray.foldRight(initial: R, operation: (Byte, acc: R) -> R):
R {\n  var index = lastIndex\n  var accumulator = initial\n  while (index >= 0) {\n    accumulator =
operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with
[initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n *
Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element
and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R>

```

```

ShortArray.foldRight(initial: R, operation: (Short, acc: R) -> R): R {
  var index = lastIndex
  var accumulator = initial
  while (index >= 0) {
    accumulator = operation(get(index--), accumulator)
  }
  return accumulator
}

```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.

Returns the specified [initial] value if the array is empty.

@param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.

```

public inline fun <R> IntArray.foldRight(initial: R, operation: (Int, acc: R) -> R): R {
  var index = lastIndex
  var accumulator = initial
  while (index >= 0) {
    accumulator = operation(get(index--), accumulator)
  }
  return accumulator
}

```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.

Returns the specified [initial] value if the array is empty.

@param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.

```

public inline fun <R> LongArray.foldRight(initial: R, operation: (Long, acc: R) -> R): R {
  var index = lastIndex
  var accumulator = initial
  while (index >= 0) {
    accumulator = operation(get(index--), accumulator)
  }
  return accumulator
}

```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.

Returns the specified [initial] value if the array is empty.

@param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.

```

public inline fun <R> FloatArray.foldRight(initial: R, operation: (Float, acc: R) -> R): R {
  var index = lastIndex
  var accumulator = initial
  while (index >= 0) {
    accumulator = operation(get(index--), accumulator)
  }
  return accumulator
}

```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.

Returns the specified [initial] value if the array is empty.

@param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.

```

public inline fun <R> DoubleArray.foldRight(initial: R, operation: (Double, acc: R) -> R): R {
  var index = lastIndex
  var accumulator = initial
  while (index >= 0) {
    accumulator = operation(get(index--), accumulator)
  }
  return accumulator
}

```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.

Returns the specified [initial] value if the array is empty.

@param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.

```

public inline fun <R> BooleanArray.foldRight(initial: R, operation: (Boolean, acc: R) -> R): R {
  var index = lastIndex
  var accumulator = initial
  while (index >= 0) {
    accumulator = operation(get(index--), accumulator)
  }
  return accumulator
}

```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.

Returns the specified [initial] value if the array is empty.

@param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.

```

public inline fun <R> CharArray.foldRight(initial: R, operation: (Char, acc: R) -> R): R {
  var index = lastIndex
  var accumulator = initial
  while (index >= 0) {
    accumulator = operation(get(index--), accumulator)
  }
  return accumulator
}

```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element with its index in the original array and current accumulator value.

Returns the specified [initial] value if the array is empty.

@param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```

public inline fun <T, R> Array<out T>.foldRightIndexed(initial: R, operation: (index: Int, T, acc: R) -> R): R {
  var index = lastIndex
  var accumulator = initial
  while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
    --index
  }
  return accumulator
}

```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element with its index in the original array and current accumulator value.

Returns the specified [initial] value if the array is empty.

@param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```

public inline fun <R> ByteArray.foldRightIndexed(initial: R, operation: (index: Int, Byte, acc: R) -> R): R {
  var index = lastIndex
  var accumulator = initial
  while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
    --index
  }
  return accumulator
}

```

```

accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to
left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the
specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an
element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n
*\npublic inline fun <R> ShortArray.foldRightIndexed(initial: R, operation: (index: Int, Short, acc: R) -> R): R {\n
var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index,
get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with
[initial] value and applying [operation] from right to left\n * to each element with its index in the original array and
current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param
[operation] function that takes the index of an element, the element itself\n * and current accumulator value, and
calculates the next accumulator value.\n *\npublic inline fun <R> IntArray.foldRightIndexed(initial: R, operation:
(index: Int, Int, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n *
Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element with
its index in the original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array
is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself\n * and current
accumulator value, and calculates the next accumulator value.\n *\npublic inline fun <R>
LongArray.foldRightIndexed(initial: R, operation: (index: Int, Long, acc: R) -> R): R {\n var index = lastIndex\n
var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n
--index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying
[operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n
*\n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the
index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator
value.\n *\npublic inline fun <R> FloatArray.foldRightIndexed(initial: R, operation: (index: Int, Float, acc: R) ->
R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator =
operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates
value starting with [initial] value and applying [operation] from right to left\n * to each element with its index in the
original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n
*\n * @param [operation] function that takes the index of an element, the element itself\n * and current accumulator
value, and calculates the next accumulator value.\n *\npublic inline fun <R> DoubleArray.foldRightIndexed(initial:
R, operation: (index: Int, Double, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while
(index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return
accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to
left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the
specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an
element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n
*\npublic inline fun <R> BooleanArray.foldRightIndexed(initial: R, operation: (index: Int, Boolean, acc: R) -> R):
R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator =
operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates
value starting with [initial] value and applying [operation] from right to left\n * to each element with its index in the
original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n
*\n * @param [operation] function that takes the index of an element, the element itself\n * and current accumulator
value, and calculates the next accumulator value.\n *\npublic inline fun <R> CharArray.foldRightIndexed(initial: R,
operation: (index: Int, Char, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while
(index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return
accumulator\n}\n\n/**\n * Performs the given [action] on each element.\n *\npublic inline fun <T> Array<out
T>.forEach(action: (T) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n * Performs the given
[action] on each element.\n *\npublic inline fun ByteArray.forEach(action: (Byte) -> Unit): Unit {\n for (element

```



```

in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element.\n */\npublic inline fun
ShortArray.forEach(action: (Short) -> Unit): Unit {\n    for (element in this) action(element)\n}\n\n/**\n * Performs
the given [action] on each element.\n */\npublic inline fun IntArray.forEach(action: (Int) -> Unit): Unit {\n    for
(element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element.\n */\npublic inline fun
LongArray.forEach(action: (Long) -> Unit): Unit {\n    for (element in this) action(element)\n}\n\n/**\n * Performs
the given [action] on each element.\n */\npublic inline fun FloatArray.forEach(action: (Float) -> Unit): Unit {\n
    for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element.\n */\npublic inline
fun DoubleArray.forEach(action: (Double) -> Unit): Unit {\n    for (element in this) action(element)\n}\n\n/**\n *
Performs the given [action] on each element.\n */\npublic inline fun BooleanArray.forEach(action: (Boolean) ->
Unit): Unit {\n    for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element.\n
*/\npublic inline fun CharArray.forEach(action: (Char) -> Unit): Unit {\n    for (element in this)
action(element)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the
element.\n */\n @param [action] function that takes the index of an element and the element itself\n * and performs the
action on the element.\n */\npublic inline fun <T> Array<out T>.forEachIndexed(action: (index: Int, T) -> Unit):
Unit {\n    var index = 0\n    for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on
each element, providing sequential index with the element.\n */\n @param [action] function that takes the index of an
element and the element itself\n * and performs the action on the element.\n */\npublic inline fun
ByteArray.forEachIndexed(action: (index: Int, Byte) -> Unit): Unit {\n    var index = 0\n    for (item in this)
action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with
the element.\n */\n @param [action] function that takes the index of an element and the element itself\n * and performs
the action on the element.\n */\npublic inline fun ShortArray.forEachIndexed(action: (index: Int, Short) -> Unit):
Unit {\n    var index = 0\n    for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on
each element, providing sequential index with the element.\n */\n @param [action] function that takes the index of an
element and the element itself\n * and performs the action on the element.\n */\npublic inline fun
IntArray.forEachIndexed(action: (index: Int, Int) -> Unit): Unit {\n    var index = 0\n    for (item in this)
action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with
the element.\n */\n @param [action] function that takes the index of an element and the element itself\n * and performs
the action on the element.\n */\npublic inline fun LongArray.forEachIndexed(action: (index: Int, Long) -> Unit):
Unit {\n    var index = 0\n    for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on
each element, providing sequential index with the element.\n */\n @param [action] function that takes the index of an
element and the element itself\n * and performs the action on the element.\n */\npublic inline fun
FloatArray.forEachIndexed(action: (index: Int, Float) -> Unit): Unit {\n    var index = 0\n    for (item in this)
action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with
the element.\n */\n @param [action] function that takes the index of an element and the element itself\n * and performs
the action on the element.\n */\npublic inline fun DoubleArray.forEachIndexed(action: (index: Int, Double) -> Unit):
Unit {\n    var index = 0\n    for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on
each element, providing sequential index with the element.\n */\n @param [action] function that takes the index of an
element and the element itself\n * and performs the action on the element.\n */\npublic inline fun
BooleanArray.forEachIndexed(action: (index: Int, Boolean) -> Unit): Unit {\n    var index = 0\n    for (item in this)
action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with
the element.\n */\n @param [action] function that takes the index of an element and the element itself\n * and performs
the action on the element.\n */\npublic inline fun CharArray.forEachIndexed(action: (index: Int, Char) -> Unit): Unit
{\n    var index = 0\n    for (item in this) action(index++, item)\n}\n\n/**\n * Returns the largest element.\n */\n * \n * If
any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n @SinceKotlin("1.7")\n @kotlin.jvm.JvmName("maxOrNull")\n @Suppress("CONFLICTING_OVERLOADS")\npublic fun Array<out Double>.max(): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var
max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        max = maxOf(max, e)\n    }\n    return
max\n}\n\n/**\n * Returns the largest element.\n */\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws

```

NoSuchElementException if the array is empty.\n

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun Array<out Float>.max(): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var\n    max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        max = maxOf(max, e)\n    }\n    return\n    max\n}\n\n/**\n * Returns the largest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */
```

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun <T : Comparable<T>> Array<out T>.max(): T {\n    if (isEmpty()) throw\n    NoSuchElementException()\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (max < e)\n            max = e\n    }\n    return max\n}\n\n/**\n * Returns the largest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */
```

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun ByteArray.max(): Byte {\n    if (isEmpty()) throw NoSuchElementException()\n    var max =\n    this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (max < e) max = e\n    }\n    return max\n}\n\n/**\n * Returns the largest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */
```

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun ShortArray.max(): Short {\n    if (isEmpty()) throw NoSuchElementException()\n    var max =\n    this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (max < e) max = e\n    }\n    return max\n}\n\n/**\n * Returns the largest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */
```

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun IntArray.max(): Int {\n    if (isEmpty()) throw NoSuchElementException()\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (max < e) max = e\n    }\n    return max\n}\n\n/**\n * Returns\n * the largest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */
```

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun LongArray.max(): Long {\n    if (isEmpty()) throw NoSuchElementException()\n    var max =\n    this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (max < e) max = e\n    }\n    return max\n}\n\n/**\n * Returns the largest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws
```

NoSuchElementException if the array is empty.\n

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun FloatArray.max(): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var max =\n    this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        max = maxOf(max, e)\n    }\n    return max\n}\n\n/**\n * Returns the largest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws
```

NoSuchElementException if the array is empty.\n

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun DoubleArray.max(): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var max =\n    this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        max = maxOf(max, e)\n    }\n    return max\n}\n\n/**\n * Returns the largest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */
```

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun CharArray.max(): Char {\n    if (isEmpty()) throw NoSuchElementException()\n    var max =\n    this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (max < e) max = e\n    }\n    return max\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n * @throws
```

NoSuchElementException if the array is empty.\n * \n * @sample

samples.collections.Collections.Aggregates.maxBy\n

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <T, R : Comparable<R>> Array<out T>.maxBy(selector: (T) -> R): T {\n    if\n    (isEmpty()) throw NoSuchElementException()\n    var maxElem = this[0]\n    val lastIndex = this.lastIndex\n    if\n    (lastIndex == 0) return maxElem\n    var maxValue = selector(maxElem)\n    for (i in 1..lastIndex) {\n        val e =\n        this[i]\n        val v = selector(e)\n        if (maxValue < v) {\n            maxElem = e\n            maxValue = v\n        }\n    }\n    return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n */
```

```

* @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.maxBy\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <R : Comparable<R>> ByteArray.maxBy(selector: (Byte) -> R): Byte {\n if
(isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n
* @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.maxBy\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <R : Comparable<R>> ShortArray.maxBy(selector: (Short) -> R): Short {\n if
(isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n
* @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.maxBy\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <R : Comparable<R>> IntArray.maxBy(selector: (Int) -> R): Int {\n if (isEmpty())
throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex ==
0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val
v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return
maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n * @throws
NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.maxBy\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <R : Comparable<R>> LongArray.maxBy(selector: (Long) -> R): Long {\n if
(isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n
* @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.maxBy\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <R : Comparable<R>> FloatArray.maxBy(selector: (Float) -> R): Float {\n if
(isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n
* @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.maxBy\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <R : Comparable<R>> DoubleArray.maxBy(selector: (Double) -> R): Double {\n if
(isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n

```

```

* @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.maxBy\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow")\n@Suppress("CONFLICTING_OVERL
OADS")\npublic inline fun <R : Comparable<R>> BooleanArray.maxBy(selector: (Boolean) -> R): Boolean {\n
if (isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n
}\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n
*\n * @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.maxBy\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow")\n@Suppress("CONFLICTING_OVERL
OADS")\npublic inline fun <R : Comparable<R>> CharArray.maxBy(selector: (Char) -> R): Char {\n if
(isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n
}\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or
`null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n
*\n@SinceKotlin("1.4")\npublic inline fun <T, R : Comparable<R>> Array<out T>.maxByOrNull(selector: (T) ->
R): T? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex
== 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n
val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return
maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are
no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n
*\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> ByteArray.maxByOrNull(selector: (Byte) ->
R): Byte? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n
}\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or
`null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n
*\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> ShortArray.maxByOrNull(selector: (Short) ->
R): Short? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n
}\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or
`null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n
*\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> IntArray.maxByOrNull(selector: (Int) -> R):
Int? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex ==
0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val
v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return
maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are
no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n
*\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> LongArray.maxByOrNull(selector: (Long) ->
R): Long? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n
}\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or
`null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n
*\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> FloatArray.maxByOrNull(selector: (Float) ->

```

```

R): Float? {\n  if (isEmpty()) return null\n  var maxElem = this[0]\n  val lastIndex = this.lastIndex\n  if
(lastIndex == 0) return maxElem\n  var maxValue = selector(maxElem)\n  for (i in 1..lastIndex) {\n    val e =
this[i]\n    val v = selector(e)\n    if (maxValue < v) {\n      maxElem = e\n      maxValue = v\n    }\n  }\n  return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or
`null` if there are no elements.\n */\n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n
*/\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> DoubleArray.maxByOrNull(selector: (Double)
-> R): Double? {\n  if (isEmpty()) return null\n  var maxElem = this[0]\n  val lastIndex = this.lastIndex\n  if
(lastIndex == 0) return maxElem\n  var maxValue = selector(maxElem)\n  for (i in 1..lastIndex) {\n    val e =
this[i]\n    val v = selector(e)\n    if (maxValue < v) {\n      maxElem = e\n      maxValue = v\n    }\n  }\n  return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or
`null` if there are no elements.\n */\n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n
*/\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> BooleanArray.maxByOrNull(selector:
(Boolean) -> R): Boolean? {\n  if (isEmpty()) return null\n  var maxElem = this[0]\n  val lastIndex =
this.lastIndex\n  if (lastIndex == 0) return maxElem\n  var maxValue = selector(maxElem)\n  for (i in
1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if (maxValue < v) {\n      maxElem = e\n
maxValue = v\n    }\n  }\n  return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value
of the given function or `null` if there are no elements.\n */\n * @sample
samples.collections.Collections.Aggregates.maxByOrNull\n
*/\n@SinceKotlin("1.4")\npublic inline fun <R :
Comparable<R>> CharArray.maxByOrNull(selector: (Char) -> R): Char? {\n  if (isEmpty()) return null\n  var
maxElem = this[0]\n  val lastIndex = this.lastIndex\n  if (lastIndex == 0) return maxElem\n  var maxValue =
selector(maxElem)\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if (maxValue < v)
{\n      maxElem = e\n      maxValue = v\n    }\n  }\n  return maxElem\n}\n\n/**\n * Returns the largest
value among all values produced by [selector] function\n * applied to each element in the array.\n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * @throws
NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.maxOf(selector: (T) ->
Double): Double {\n  if (isEmpty()) throw NoSuchElementException()\n  var maxValue = selector(this[0])\n  for
(i in 1..lastIndex) {\n    val v = selector(this[i])\n    maxValue = maxOf(maxValue, v)\n  }\n  return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.maxOf(selector: (Byte) ->
Double): Double {\n  if (isEmpty()) throw NoSuchElementException()\n  var maxValue = selector(this[0])\n  for
(i in 1..lastIndex) {\n    val v = selector(this[i])\n    maxValue = maxOf(maxValue, v)\n  }\n  return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.maxOf(selector: (Short) ->
Double): Double {\n  if (isEmpty()) throw NoSuchElementException()\n  var maxValue = selector(this[0])\n  for
(i in 1..lastIndex) {\n    val v = selector(this[i])\n    maxValue = maxOf(maxValue, v)\n  }\n  return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.maxOf(selector: (Int) -> Double):

```

```

Double {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return
    maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun LongArray.maxOf(selector: (Long) ->
Double): Double {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return
    maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun FloatArray.maxOf(selector: (Float) ->
Double): Double {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return
    maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun DoubleArray.maxOf(selector: (Double) ->
Double): Double {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return
    maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun BooleanArray.maxOf(selector: (Boolean) ->
Double): Double {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return
    maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun CharArray.maxOf(selector: (Char) ->
Double): Double {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return
    maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.maxOf(selector: (T) ->
Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return
    maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution

```

```

ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.maxOf(selector: (Byte) -> Float):
Float {\n  if (isEmpty()) throw NoSuchElementException()\n  var maxValue = selector(this[0])\n  for (i in
1..lastIndex) {\n    val v = selector(this[i])\n    maxValue = maxOf(maxValue, v)\n  }\n  return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.maxOf(selector: (Short) ->
Float): Float {\n  if (isEmpty()) throw NoSuchElementException()\n  var maxValue = selector(this[0])\n  for (i
in 1..lastIndex) {\n    val v = selector(this[i])\n    maxValue = maxOf(maxValue, v)\n  }\n  return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.maxOf(selector: (Int) -> Float):
Float {\n  if (isEmpty()) throw NoSuchElementException()\n  var maxValue = selector(this[0])\n  for (i in
1..lastIndex) {\n    val v = selector(this[i])\n    maxValue = maxOf(maxValue, v)\n  }\n  return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.maxOf(selector: (Long) ->
Float): Float {\n  if (isEmpty()) throw NoSuchElementException()\n  var maxValue = selector(this[0])\n  for (i
in 1..lastIndex) {\n    val v = selector(this[i])\n    maxValue = maxOf(maxValue, v)\n  }\n  return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.maxOf(selector: (Float) ->
Float): Float {\n  if (isEmpty()) throw NoSuchElementException()\n  var maxValue = selector(this[0])\n  for (i
in 1..lastIndex) {\n    val v = selector(this[i])\n    maxValue = maxOf(maxValue, v)\n  }\n  return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.maxOf(selector: (Double) ->
Float): Float {\n  if (isEmpty()) throw NoSuchElementException()\n  var maxValue = selector(this[0])\n  for (i
in 1..lastIndex) {\n    val v = selector(this[i])\n    maxValue = maxOf(maxValue, v)\n  }\n  return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.maxOf(selector: (Boolean) ->
Float): Float {\n  if (isEmpty()) throw NoSuchElementException()\n  var maxValue = selector(this[0])\n  for (i
in 1..lastIndex) {\n    val v = selector(this[i])\n    maxValue = maxOf(maxValue, v)\n  }\n  return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.maxOf(selector: (Char) -> Float):
Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in
1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Array<out
T>.maxOf(selector: (T) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue =
selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n
maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the
array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
ByteArray.maxOf(selector: (Byte) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var
maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n
maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
ShortArray.maxOf(selector: (Short) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var
maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n
maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
IntArray.maxOf(selector: (Int) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue =
selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n
maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the
array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
LongArray.maxOf(selector: (Long) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var
maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n
maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
FloatArray.maxOf(selector: (Float) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var
maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n
maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n

```



```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
DoubleArray.maxOf(selector: (Double) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n
maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
BooleanArray.maxOf(selector: (Boolean) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n
maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
CharArray.maxOf(selector: (Char) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n
maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If
any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.maxOfOrNull(selector:
(T) -> Double): Double? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function
is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.maxOfOrNull(selector: (Byte) ->
Double): Double? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n
val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns
the largest value among all values produced by [selector] function\n * applied to each element in the array or `null`
if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.maxOfOrNull(selector: (Short) -
> Double): Double? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex)
{\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n *
Returns the largest value among all values produced by [selector] function\n * applied to each element in the array
or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.maxOfOrNull(selector: (Int) ->
Double): Double? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n
val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns
the largest value among all values produced by [selector] function\n * applied to each element in the array or `null`
if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is

```

```

`NaN`.
*/@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\npublic inline fun LongArray.maxOrNull(selector: (Long) -
> Double): Double? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex)
{\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue}\n\n/**\n *
Returns the largest value among all values produced by [selector] function\n * applied to each element in the array
or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n
*/@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\npublic inline fun FloatArray.maxOrNull(selector: (Float) -
> Double): Double? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex)
{\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue}\n\n/**\n *
Returns the largest value among all values produced by [selector] function\n * applied to each element in the array
or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n
*/@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.maxOrNull(selector:
(Double) -> Double): Double? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return
maxValue}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function
is `NaN`, the returned result is `NaN`.\n
*/@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.maxOrNull(selector:
(Boolean) -> Double): Double? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return
maxValue}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function
is `NaN`, the returned result is `NaN`.\n
*/@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\npublic inline fun CharArray.maxOrNull(selector: (Char) ->
Double): Double? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n
val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue}\n\n/**\n * Returns
the largest value among all values produced by [selector] function\n * applied to each element in the array or `null`
if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n
*/@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.maxOrNull(selector:
(T) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex)
{\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue}\n\n/**\n *
Returns the largest value among all values produced by [selector] function\n * applied to each element in the array
or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n
*/@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\npublic inline fun ByteArray.maxOrNull(selector: (Byte) ->
Float): Float? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n
val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue}\n\n/**\n * Returns
the largest value among all values produced by [selector] function\n * applied to each element in the array or `null`

```

if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.maxOrNull(selector: (Short) -> Float): Float? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.maxOrNull(selector: (Int) -> Float): Float? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.maxOrNull(selector: (Long) -> Float): Float? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.maxOrNull(selector: (Float) -> Float): Float? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.maxOrNull(selector: (Double) -> Float): Float? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.maxOrNull(selector: (Boolean) -> Float): Float? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.maxOrNull(selector: (Char) -> Float): Float? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns
```

the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Array<out T>.maxOrNull(selector: (T) -> R): R? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nByteArray.maxOrNull(selector: (Byte) -> R): R? {\n    if (isEmpty()) return null\n    var maxValue =\n    selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n * Returns the largest value among all values produced\n by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nShortArray.maxOrNull(selector: (Short) -> R): R? {\n    if (isEmpty()) return null\n    var maxValue =\n    selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n * Returns the largest value among all values produced\n by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nIntArray.maxOrNull(selector: (Int) -> R): R? {\n    if (isEmpty()) return null\n    var maxValue =\n    selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n * Returns the largest value among all values produced\n by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nLongArray.maxOrNull(selector: (Long) -> R): R? {\n    if (isEmpty()) return null\n    var maxValue =\n    selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n * Returns the largest value among all values produced\n by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nFloatArray.maxOrNull(selector: (Float) -> R): R? {\n    if (isEmpty()) return null\n    var maxValue =\n    selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n * Returns the largest value among all values produced\n by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nDoubleArray.maxOrNull(selector: (Double) -> R): R? {\n    if (isEmpty()) return null\n    var maxValue =\n    selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n * Returns the largest value among all values produced\n by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nBooleanArray.maxOrNull(selector: (Boolean) -> R): R? {\n    if (isEmpty()) return null\n    var maxValue =\n    selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n
```

```

maxValue = v\n    }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
CharArray.maxOrNull(selector: (Char) -> R): R? {\n    if (isEmpty()) return null\n    var maxValue =
selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n
maxValue = v\n    }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n *
@throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Array<out
T>.maxOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n    if (isEmpty()) throw
NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =
selector(this[i])\n        if (comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return
maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if
the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ByteArray.maxOfWith(comparator:
Comparator<in R>, selector: (Byte) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var
maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ShortArray.maxOfWith(comparator:
Comparator<in R>, selector: (Short) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var
maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> IntArray.maxOfWith(comparator:
Comparator<in R>, selector: (Int) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue
= selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.maxOfWith(comparator:
Comparator<in R>, selector: (Long) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var
maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> FloatArray.maxOfWith(comparator:
Comparator<in R>, selector: (Float) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var

```

```

maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n * @SinceKotlin("1.4")\n * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n * @OverloadResolution
ByLambdaReturnType\n * @kotlin.internal.InlineOnly\n * public inline fun <R> DoubleArray.maxOfWith(comparator:
Comparator<in R>, selector: (Double) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var
maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n * @SinceKotlin("1.4")\n * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n * @OverloadResolution
ByLambdaReturnType\n * @kotlin.internal.InlineOnly\n * public inline fun <R> BooleanArray.maxOfWith(comparator:
Comparator<in R>, selector: (Boolean) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var
maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n * @SinceKotlin("1.4")\n * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n * @OverloadResolution
ByLambdaReturnType\n * @kotlin.internal.InlineOnly\n * public inline fun <R> CharArray.maxOfWith(comparator:
Comparator<in R>, selector: (Char) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var
maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n * @SinceKotlin("1.4")\n * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n * @OverloadResolution
ByLambdaReturnType\n * @kotlin.internal.InlineOnly\n * public inline fun <T, R> Array<out
T>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n  if (isEmpty()) return null\n
var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n * @SinceKotlin("1.4")\n * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n * @OverloadResolution
ByLambdaReturnType\n * @kotlin.internal.InlineOnly\n * public inline fun <R>
ByteArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Byte) -> R): R? {\n  if (isEmpty()) return
null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n * @SinceKotlin("1.4")\n * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n * @OverloadResolution
ByLambdaReturnType\n * @kotlin.internal.InlineOnly\n * public inline fun <R>
ShortArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Short) -> R): R? {\n  if (isEmpty())
return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n * @SinceKotlin("1.4")\n * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n * @OverloadResolution
ByLambdaReturnType\n * @kotlin.internal.InlineOnly\n * public inline fun <R>

```

```

IntArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Int) -> R): R? {\n  if (isEmpty()) return
null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
LongArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Long) -> R): R? {\n  if (isEmpty())
return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
FloatArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Float) -> R): R? {\n  if (isEmpty())
return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
DoubleArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Double) -> R): R? {\n  if (isEmpty())
return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
BooleanArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Boolean) -> R): R? {\n  if (isEmpty())
return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
CharArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {\n  if (isEmpty()) return
null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n
*\n@SinceKotlin("1.4")\npublic fun Array<out Double>.maxOrNull(): Double? {\n  if (isEmpty()) return null\n
var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    max = maxOf(max, e)\n  }\n  return
max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n * \n * If any of elements is `NaN`
returns `NaN`.\n
*\n@SinceKotlin("1.4")\npublic fun Array<out Float>.maxOrNull(): Float? {\n  if (isEmpty())
return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    max = maxOf(max, e)\n  }\n
return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\npublic fun <T : Comparable<T>> Array<out T>.maxOrNull(): T? {\n  if (isEmpty())
return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (max < e) max = e\n  }\n
return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n

```

```

*^@SinceKotlin("1.4")\npublic fun ByteArray.maxOrNull(): Byte? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (max < e) max = e\n  }\n  return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n */\n*^@SinceKotlin("1.4")\npublic fun ShortArray.maxOrNull(): Short? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (max < e) max = e\n  }\n  return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n */\n*^@SinceKotlin("1.4")\npublic fun IntArray.maxOrNull(): Int? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (max < e) max = e\n  }\n  return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n */\n*^@SinceKotlin("1.4")\npublic fun LongArray.maxOrNull(): Long? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (max < e) max = e\n  }\n  return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n*^@SinceKotlin("1.4")\npublic fun FloatArray.maxOrNull(): Float? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    max = maxOf(max, e)\n  }\n  return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n*^@SinceKotlin("1.4")\npublic fun DoubleArray.maxOrNull(): Double? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    max = maxOf(max, e)\n  }\n  return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n */\n*^@SinceKotlin("1.4")\npublic fun CharArray.maxOrNull(): Char? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (max < e) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */\n*^@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun <T> Array<out T>.maxWith(comparator: Comparator<in T>): T {\n  if (isEmpty()) throw NoSuchElementException()\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */\n*^@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun ByteArray.maxWith(comparator: Comparator<in Byte>): Byte {\n  if (isEmpty()) throw NoSuchElementException()\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */\n*^@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun ShortArray.maxWith(comparator: Comparator<in Short>): Short {\n  if (isEmpty()) throw NoSuchElementException()\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */\n*^@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun IntArray.maxWith(comparator: Comparator<in Int>): Int {\n  if (isEmpty()) throw NoSuchElementException()\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */\n*^@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun LongArray.maxWith(comparator: Comparator<in Long>): Long {\n  if (isEmpty()) throw NoSuchElementException()\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if

```



```

(comparator.compare(max, e) < 0) max = e\n } \n return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is
empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow")\n@Suppress("CONFLICTING_OVER
LOADS")\npublic fun FloatArray.maxWith(comparator: Comparator<in Float>): Float {\n if (isEmpty()) throw
NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(max, e) < 0) max = e\n } \n return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is
empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow")\n@Suppress("CONFLICTING_OVER
LOADS")\npublic fun DoubleArray.maxWith(comparator: Comparator<in Double>): Double {\n if (isEmpty())
throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(max, e) < 0) max = e\n } \n return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is
empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow")\n@Suppress("CONFLICTING_OVER
LOADS")\npublic fun BooleanArray.maxWith(comparator: Comparator<in Boolean>): Boolean {\n if
(isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e =
this[i]\n if (comparator.compare(max, e) < 0) max = e\n } \n return max\n}\n\n/**\n * Returns the first
element having the largest value according to the provided [comparator].\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow")\n@Suppress("CONFLICTING_OVER
LOADS")\npublic fun CharArray.maxWith(comparator: Comparator<in Char>): Char {\n if (isEmpty()) throw
NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(max, e) < 0) max = e\n } \n return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\npublic fun <T> Array<out T>.maxWithOrNull(comparator: Comparator<in T>): T? {\n
if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(max, e) < 0) max = e\n } \n return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\npublic fun ByteArray.maxWithOrNull(comparator: Comparator<in Byte>): Byte? {\n
if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(max, e) < 0) max = e\n } \n return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\npublic fun ShortArray.maxWithOrNull(comparator: Comparator<in Short>): Short? {\n
if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(max, e) < 0) max = e\n } \n return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\npublic fun IntArray.maxWithOrNull(comparator: Comparator<in Int>): Int? {\n if
(isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(max, e) < 0) max = e\n } \n return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\npublic fun LongArray.maxWithOrNull(comparator: Comparator<in Long>): Long? {\n
if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(max, e) < 0) max = e\n } \n return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\npublic fun FloatArray.maxWithOrNull(comparator: Comparator<in Float>): Float? {\n
if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if

```

```

(comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator] or `null` if there are no elements.\n
*\n\n@SinceKotlin("1.4")\npublic fun DoubleArray.maxWithOrNull(comparator: Comparator<in Double>):
Double? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator] or `null` if there are no elements.\n
*\n\n@SinceKotlin("1.4")\npublic fun BooleanArray.maxWithOrNull(comparator: Comparator<in Boolean>):
Boolean? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator] or `null` if there are no elements.\n
*\n\n@SinceKotlin("1.4")\npublic fun CharArray.maxWithOrNull(comparator: Comparator<in Char>): Char? {\n
if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if
(comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the smallest element.\n * \n *
If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOA
DS")\npublic fun Array<out Double>.min(): Double {\n  if (isEmpty()) throw NoSuchElementException()\n  var
min = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    min = minOf(min, e)\n  }\n  return
min\n}\n\n/**\n * Returns the smallest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n *
@throws NoSuchElementException if the array is empty.\n
*\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOA
DS")\npublic fun Array<out Float>.min(): Float {\n  if (isEmpty()) throw NoSuchElementException()\n  var min
= this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    min = minOf(min, e)\n  }\n  return min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOA
DS")\npublic fun <T : Comparable<T>> Array<out T>.min(): T {\n  if (isEmpty()) throw
NoSuchElementException()\n  var min = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (min > e)
min = e\n  }\n  return min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOA
DS")\npublic fun ByteArray.min(): Byte {\n  if (isEmpty()) throw NoSuchElementException()\n  var min =
this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (min > e) min = e\n  }\n  return min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOA
DS")\npublic fun ShortArray.min(): Short {\n  if (isEmpty()) throw NoSuchElementException()\n  var min =
this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (min > e) min = e\n  }\n  return min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOA
DS")\npublic fun IntArray.min(): Int {\n  if (isEmpty()) throw NoSuchElementException()\n  var min = this[0]\n
for (i in 1..lastIndex) {\n    val e = this[i]\n    if (min > e) min = e\n  }\n  return min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOA
DS")\npublic fun LongArray.min(): Long {\n  if (isEmpty()) throw NoSuchElementException()\n  var min =
this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (min > e) min = e\n  }\n  return min\n}\n\n/**\n * Returns the smallest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOA
DS")\npublic fun FloatArray.min(): Float {\n  if (isEmpty()) throw NoSuchElementException()\n  var min =
this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    min = minOf(min, e)\n  }\n  return min\n}\n\n/**\n *

```

Returns the smallest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n

```

*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun DoubleArray.min(): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        min = minOf(min, e)\n    }\n    return min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun CharArray.min(): Char {\n    if (isEmpty()) throw NoSuchElementException()\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return min\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample\n * samples.collections.Collections.Aggregates.minBy\n */\n*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <T, R : Comparable<R>> Array<out T>.minBy(selector: (T) -> R): T {\n    if (isEmpty())\n        throw NoSuchElementException()\n    var minElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0)\n        return minElem\n    var minValue = selector(minElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (minValue > v) {\n            minElem = e\n            minValue = v\n        }\n    }\n    return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample\n * samples.collections.Collections.Aggregates.minBy\n */\n*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <R : Comparable<R>> ByteArray.minBy(selector: (Byte) -> R): Byte {\n    if (isEmpty())\n        throw NoSuchElementException()\n    var minElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0)\n        return minElem\n    var minValue = selector(minElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (minValue > v) {\n            minElem = e\n            minValue = v\n        }\n    }\n    return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample\n * samples.collections.Collections.Aggregates.minBy\n */\n*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <R : Comparable<R>> ShortArray.minBy(selector: (Short) -> R): Short {\n    if (isEmpty())\n        throw NoSuchElementException()\n    var minElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0)\n        return minElem\n    var minValue = selector(minElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (minValue > v) {\n            minElem = e\n            minValue = v\n        }\n    }\n    return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample\n * samples.collections.Collections.Aggregates.minBy\n */\n*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <R : Comparable<R>> IntArray.minBy(selector: (Int) -> R): Int {\n    if (isEmpty())\n        throw NoSuchElementException()\n    var minElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0)\n        return minElem\n    var minValue = selector(minElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (minValue > v) {\n            minElem = e\n            minValue = v\n        }\n    }\n    return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample\n * samples.collections.Collections.Aggregates.minBy\n */\n*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <R : Comparable<R>> LongArray.minBy(selector: (Long) -> R): Long {\n    if (isEmpty())\n        throw NoSuchElementException()\n    var minElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0)\n        return minElem\n    var minValue = selector(minElem)\n    for (i in 1..lastIndex) {\n        val e =

```

```

this[i]\n    val v = selector(e)\n    if (minValue > v) {\n        minElem = e\n        minValue = v\n    }\n}\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.minBy\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <R : Comparable<R>> FloatArray.minBy(selector: (Float) -> R): Float {\n    if
(isEmpty()) throw NoSuchElementException()\n    var minElem = this[0]\n    val lastIndex = this.lastIndex\n    if
(lastIndex == 0) return minElem\n    var minValue = selector(minElem)\n    for (i in 1..lastIndex) {\n        val e =
this[i]\n        val v = selector(e)\n        if (minValue > v) {\n            minElem = e\n            minValue = v\n        }\n    }\n    return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.minBy\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <R : Comparable<R>> DoubleArray.minBy(selector: (Double) -> R): Double {\n    if
(isEmpty()) throw NoSuchElementException()\n    var minElem = this[0]\n    val lastIndex = this.lastIndex\n    if
(lastIndex == 0) return minElem\n    var minValue = selector(minElem)\n    for (i in 1..lastIndex) {\n        val e =
this[i]\n        val v = selector(e)\n        if (minValue > v) {\n            minElem = e\n            minValue = v\n        }\n    }\n    return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.minBy\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun <R : Comparable<R>> BooleanArray.minBy(selector: (Boolean) -> R): Boolean {\n    if
(isEmpty()) throw NoSuchElementException()\n    var minElem = this[0]\n    val lastIndex = this.lastIndex\n    if
(lastIndex == 0) return minElem\n    var minValue = selector(minElem)\n    for (i in 1..lastIndex) {\n        val e =
this[i]\n        val v = selector(e)\n        if (minValue > v) {\n            minElem = e\n            minValue = v\n        }\n    }\n    return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*\n@SinceKotlin("1.4")\npublic inline fun <T, R : Comparable<R>> Array<out T>.minByOrNull(selector: (T) ->
R): T? {\n    if (isEmpty()) return null\n    var minElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex
== 0) return minElem\n    var minValue = selector(minElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (minValue > v) {\n            minElem = e\n            minValue = v\n        }\n    }\n    return
minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> ByteArray.minByOrNull(selector: (Byte) ->
R): Byte? {\n    if (isEmpty()) return null\n    var minElem = this[0]\n    val lastIndex = this.lastIndex\n    if
(lastIndex == 0) return minElem\n    var minValue = selector(minElem)\n    for (i in 1..lastIndex) {\n        val e =
this[i]\n        val v = selector(e)\n        if (minValue > v) {\n            minElem = e\n            minValue = v\n        }\n    }\n    return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> ShortArray.minByOrNull(selector: (Short) ->

```

```

R): Short? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }}\n }
}\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or
`null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*\n*\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> IntArray.minByOrNull(selector: (Int) -> R):
Int? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex ==
0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val
v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }}\n } return
minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there
are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*\n*\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> LongArray.minByOrNull(selector: (Long) ->
R): Long? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }}\n }
}\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or
`null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*\n*\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> FloatArray.minByOrNull(selector: (Float) ->
R): Float? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }}\n }
}\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or
`null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*\n*\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> DoubleArray.minByOrNull(selector: (Double)
-> R): Double? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }}\n }
}\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or
`null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*\n*\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> BooleanArray.minByOrNull(selector:
(Boolean) -> R): Boolean? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex =
this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in
1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n
minValue = v\n }}\n } return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value
of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.minByOrNull\n *\n*\n@SinceKotlin("1.4")\npublic inline fun <R :
Comparable<R>> CharArray.minByOrNull(selector: (Char) -> R): Char? {\n if (isEmpty()) return null\n var
minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue =
selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v)
{\n minElem = e\n minValue = v\n }}\n } return minElem\n}\n\n/**\n * Returns the smallest
value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.minOf(selector: (T) ->
Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for
(i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }}\n return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to

```

each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.minOf(selector: (Byte) ->
Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for
(i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.minOf(selector: (Short) ->
Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for
(i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.minOf(selector: (Int) -> Double):
Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.minOf(selector: (Long) ->
Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for
(i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.minOf(selector: (Float) ->
Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for
(i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.minOf(selector: (Double) ->
Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for
(i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.minOf(selector: (Boolean) ->
Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for
(i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n

```

```

minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.minOf(selector: (Char) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.minOf(selector: (T) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.minOf(selector: (Byte) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.minOf(selector: (Short) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.minOf(selector: (Int) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.minOf(selector: (Long) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.minOf(selector: (Float) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in

```

```

1..lastIndex) {\n    val v = selector(this[i])\n    minValue = minOf(minValue, v)\n } \n return
minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.minOf(selector: (Double) ->
Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in
1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.minOf(selector: (Boolean) ->
Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in
1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.minOf(selector: (Char) -> Float):
Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in
1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Array<out
T>.minOf(selector: (T) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue =
selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n
minValue = v\n        }\n    }\n    return minValue}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the
array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
ByteArray.minOf(selector: (Byte) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue
= selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n
minValue = v\n        }\n    }\n    return minValue}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the
array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
ShortArray.minOf(selector: (Short) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var
minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n
minValue = v\n        }\n    }\n    return minValue}\n\n/**\n * Returns the smallest value among all values
produced by [selector] function\n * applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
IntArray.minOf(selector: (Int) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue =

```



```

selector(this[0])\n for (i in 1..lastIndex) {\n     val v = selector(this[i])\n     if (minValue > v) {\n
minValue = v\n     }\n } return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the
array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
LongArray.minOf(selector: (Long) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n     val v = selector(this[i])\n     if (minValue > v) {\n
minValue = v\n     }\n } return minValue\n}\n\n/**\n * Returns the smallest value among all values
produced by [selector] function\n * applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
FloatArray.minOf(selector: (Float) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n     val v = selector(this[i])\n     if (minValue > v) {\n
minValue = v\n     }\n } return minValue\n}\n\n/**\n * Returns the smallest value among all values
produced by [selector] function\n * applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
DoubleArray.minOf(selector: (Double) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n     val v = selector(this[i])\n     if (minValue > v) {\n
minValue = v\n     }\n } return minValue\n}\n\n/**\n * Returns the smallest value among all values
produced by [selector] function\n * applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
BooleanArray.minOf(selector: (Boolean) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n     val v = selector(this[i])\n     if (minValue > v) {\n
minValue = v\n     }\n } return minValue\n}\n\n/**\n * Returns the smallest value among all values
produced by [selector] function\n * applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
CharArray.minOf(selector: (Char) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue
= selector(this[0])\n for (i in 1..lastIndex) {\n     val v = selector(this[i])\n     if (minValue > v) {\n
minValue = v\n     }\n } return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.minOfOrNull(selector:
(T) -> Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in
1..lastIndex) {\n     val v = selector(this[i])\n     minValue = minOf(minValue, v)\n } return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function
is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.minOfOrNull(selector: (Byte) ->

```

Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.minOfOrNull(selector: (Short) -> Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.minOfOrNull(selector: (Int) -> Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.minOfOrNull(selector: (Long) -> Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.minOfOrNull(selector: (Float) -> Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.minOfOrNull(selector: (Double) -> Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.minOfOrNull(selector: (Boolean) -> Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
```

ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.minOfOrNull(selector: (Char) -> Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.minOfOrNull(selector: (T) -> Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.minOfOrNull(selector: (Byte) -> Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.minOfOrNull(selector: (Short) -> Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.minOfOrNull(selector: (Int) -> Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.minOfOrNull(selector: (Long) -> Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.minOfOrNull(selector: (Float) -> Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.minOfOrNull(selector:\n(Double) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in\n1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return\nminValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to\n each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function\n is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.minOfOrNull(selector:\n(Boolean) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in\n1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return\nminValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to\n each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function\n is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.minOfOrNull(selector: (Char) ->\nFloat): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the\n smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if\n there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Array<out\nT>.minOfOrNull(selector: (T) -> R): R? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for\n(i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return\nminValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * \n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nByteArray.minOfOrNull(selector: (Byte) -> R): R? {\n    if (isEmpty()) return null\n    var minValue =\nselector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return\nminValue\n}\n\n/**\n * Returns the smallest value among all values produced\n by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nShortArray.minOfOrNull(selector: (Short) -> R): R? {\n    if (isEmpty()) return null\n    var minValue =\nselector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return\nminValue\n}\n\n/**\n * Returns the smallest value among all values produced\n by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nIntArray.minOfOrNull(selector: (Int) -> R): R? {\n    if (isEmpty()) return null\n    var minValue =\nselector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return\nminValue\n}\n\n/**\n * Returns the smallest value among all values produced\n by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nLongArray.minOfOrNull(selector: (Long) -> R): R? {\n    if (isEmpty()) return null\n    var minValue =\nselector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n
```

```

minValue = v\n    }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
FloatArray.minOrNull(selector: (Float) -> R): R? {\n    if (isEmpty()) return null\n    var minValue =
selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n
minValue = v\n    }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
DoubleArray.minOrNull(selector: (Double) -> R): R? {\n    if (isEmpty()) return null\n    var minValue =
selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n
minValue = v\n    }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
BooleanArray.minOrNull(selector: (Boolean) -> R): R? {\n    if (isEmpty()) return null\n    var minValue =
selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n
minValue = v\n    }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
CharArray.minOrNull(selector: (Char) -> R): R? {\n    if (isEmpty()) return null\n    var minValue =
selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n
minValue = v\n    }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n *
@throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Array<out
T>.minOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n    if (isEmpty()) throw
NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =
selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return
minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if
the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ByteArray.minOfWith(comparator:
Comparator<in R>, selector: (Byte) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var
minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ShortArray.minOfWith(comparator:
Comparator<in R>, selector: (Short) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var
minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]

```

```

function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> IntArray.minOfWith(comparator:
Comparator<in R>, selector: (Int) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue
= selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue,
v) > 0) {\n minValue = v\n }\n }\n return minValue}\n\n/**\n * Returns the smallest value
according to the provided [comparator]\n * among all values produced by [selector] function applied to each
element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.minOfWith(comparator:
Comparator<in R>, selector: (Long) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> FloatArray.minOfWith(comparator:
Comparator<in R>, selector: (Float) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> DoubleArray.minOfWith(comparator:
Comparator<in R>, selector: (Double) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> BooleanArray.minOfWith(comparator:
Comparator<in R>, selector: (Boolean) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharArray.minOfWith(comparator:
Comparator<in R>, selector: (Char) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Array<out
T>.minOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n if (isEmpty()) return null\n
var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n

```

Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ByteArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Byte) -> R): R? {\n if (isEmpty()) return
null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/*\n *
Returns the smallest value according to the provided [comparator] among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ShortArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Short) -> R): R? {\n if (isEmpty())
return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/*\n *
Returns the smallest value according to the provided [comparator] among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
IntArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Int) -> R): R? {\n if (isEmpty()) return
null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/*\n *
Returns the smallest value according to the provided [comparator] among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
LongArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Long) -> R): R? {\n if (isEmpty()) return
null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/*\n *
Returns the smallest value according to the provided [comparator] among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
FloatArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Float) -> R): R? {\n if (isEmpty()) return
null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/*\n *
Returns the smallest value according to the provided [comparator] among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
DoubleArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Double) -> R): R? {\n if (isEmpty())
return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/*\n *
Returns the smallest value according to the provided [comparator] among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
BooleanArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Boolean) -> R): R? {\n if (isEmpty())
return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if

```

```

(comparator.compare(minValue, v) > 0) {\n        minValue = v\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]\n * function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nCharArray.minOfOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {\n    if (isEmpty()) return\n    null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if\n        (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n@SinceKotlin("1.4")\npublic fun Array<out Double>.minOrNull(): Double? {\n    if (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        min = minOf(min, e)\n    }\n    return\n    min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n * \n * If any of elements is\n * `NaN` returns `NaN`.\n */\n@SinceKotlin("1.4")\npublic fun Array<out Float>.minOrNull(): Float? {\n    if\n    (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        min = minOf(min,\n        e)\n    }\n    return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun <T : Comparable<T>> Array<out T>.minOrNull(): T? {\n    if (isEmpty())\n    return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun ByteArray.minOrNull(): Byte? {\n    if (isEmpty()) return null\n    var min =\n    this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun ShortArray.minOrNull(): Short? {\n    if (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return min\n}\n\n/**\n * Returns the smallest element or `null` if\n * there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun IntArray.minOrNull(): Int? {\n    if (isEmpty())\n    return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun LongArray.minOrNull(): Long? {\n    if (isEmpty()) return null\n    var min =\n    this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n@SinceKotlin("1.4")\npublic fun FloatArray.minOrNull(): Float? {\n    if (isEmpty()) return null\n    var min =\n    this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        min = minOf(min, e)\n    }\n    return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n@SinceKotlin("1.4")\npublic fun DoubleArray.minOrNull(): Double? {\n    if (isEmpty()) return null\n    var\n    min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        min = minOf(min, e)\n    }\n    return\n    min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun CharArray.minOrNull(): Char? {\n    if (isEmpty()) return null\n    var min =\n    this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws\n * NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVER\nLOADS")\npublic fun <T> Array<out T>.minWith(comparator: Comparator<in T>): T {\n    if (isEmpty()) throw\n    NoSuchElementException()\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if\n        (comparator.compare(min, e) > 0) min = e\n    }\n    return min\n}\n\n/**\n * Returns the first element having the\n * smallest value according to the provided [comparator].\n * \n * @throws\n * NoSuchElementException if the array is\n * empty.\n */\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVER\nLOADS")\npublic fun ByteArray.minWith(comparator: Comparator<in Byte>): Byte {\n    if (isEmpty()) throw\n    NoSuchElementException()\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if

```


(comparator.compare(min, e) > 0) min = e\n } \n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun ShortArray.minWith(comparator: Comparator<in Short>): Short {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun IntArray.minWith(comparator: Comparator<in Int>): Int {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun LongArray.minWith(comparator: Comparator<in Long>): Long {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun FloatArray.minWith(comparator: Comparator<in Float>): Float {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun DoubleArray.minWith(comparator: Comparator<in Double>): Double {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun BooleanArray.minWith(comparator: Comparator<in Boolean>): Boolean {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun CharArray.minWith(comparator: Comparator<in Char>): Char {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\npublic fun <T> Array<out T>.minWithOrNull(comparator: Comparator<in T>): T? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n
```

```

*\n@SinceKotlin("1.4")\npublic fun ByteArray.minWithOrNull(comparator: Comparator<in Byte>): Byte? {\n
if (isEmpty()) return null\n  var min = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if
(comparator.compare(min, e) > 0) min = e\n  }\n  return min\n}\n\n/**\n * Returns the first element having the
smallest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\npublic fun ShortArray.minWithOrNull(comparator: Comparator<in Short>): Short? {\n
if (isEmpty()) return null\n  var min = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if
(comparator.compare(min, e) > 0) min = e\n  }\n  return min\n}\n\n/**\n * Returns the first element having the
smallest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\npublic fun IntArray.minWithOrNull(comparator: Comparator<in Int>): Int? {\n  if
(isEmpty()) return null\n  var min = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if
(comparator.compare(min, e) > 0) min = e\n  }\n  return min\n}\n\n/**\n * Returns the first element having the
smallest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\npublic fun LongArray.minWithOrNull(comparator: Comparator<in Long>): Long? {\n
if (isEmpty()) return null\n  var min = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if
(comparator.compare(min, e) > 0) min = e\n  }\n  return min\n}\n\n/**\n * Returns the first element having the
smallest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\npublic fun FloatArray.minWithOrNull(comparator: Comparator<in Float>): Float? {\n
if (isEmpty()) return null\n  var min = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if
(comparator.compare(min, e) > 0) min = e\n  }\n  return min\n}\n\n/**\n * Returns the first element having the
smallest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\npublic fun DoubleArray.minWithOrNull(comparator: Comparator<in Double>):
Double? {\n  if (isEmpty()) return null\n  var min = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if
(comparator.compare(min, e) > 0) min = e\n  }\n  return min\n}\n\n/**\n * Returns the first element having the
smallest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\npublic fun BooleanArray.minWithOrNull(comparator: Comparator<in Boolean>):
Boolean? {\n  if (isEmpty()) return null\n  var min = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n
if (comparator.compare(min, e) > 0) min = e\n  }\n  return min\n}\n\n/**\n * Returns the first element having the
smallest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\npublic fun CharArray.minWithOrNull(comparator: Comparator<in Char>): Char? {\n
if (isEmpty()) return null\n  var min = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if
(comparator.compare(min, e) > 0) min = e\n  }\n  return min\n}\n\n/**\n * Returns `true` if the array has no
elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n *\npublic fun <T> Array<out
T>.none(): Boolean {\n  return isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n *
@sample samples.collections.Collections.Aggregates.none\n *\npublic fun ByteArray.none(): Boolean {\n  return
isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.none\n *\npublic fun ShortArray.none(): Boolean {\n  return
isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.none\n *\npublic fun IntArray.none(): Boolean {\n  return
isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.none\n *\npublic fun LongArray.none(): Boolean {\n  return
isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.none\n *\npublic fun FloatArray.none(): Boolean {\n  return
isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.none\n *\npublic fun DoubleArray.none(): Boolean {\n  return
isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.none\n *\npublic fun BooleanArray.none(): Boolean {\n  return
isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.none\n *\npublic fun CharArray.none(): Boolean {\n  return

```

```

isEmpty()\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n *^npublic inline fun <T> Array<out
T>.none(predicate: (T) -> Boolean): Boolean {\n  for (element in this) if (predicate(element)) return false\n  return
true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n *^npublic inline fun ByteArray.none(predicate:
(Byte) -> Boolean): Boolean {\n  for (element in this) if (predicate(element)) return false\n  return
true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n *^npublic inline fun ShortArray.none(predicate:
(Short) -> Boolean): Boolean {\n  for (element in this) if (predicate(element)) return false\n  return
true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n *^npublic inline fun IntArray.none(predicate: (Int)
-> Boolean): Boolean {\n  for (element in this) if (predicate(element)) return false\n  return true\n}\n\n/**\n *
Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n *^npublic inline fun LongArray.none(predicate:
(Long) -> Boolean): Boolean {\n  for (element in this) if (predicate(element)) return false\n  return
true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n *^npublic inline fun FloatArray.none(predicate:
(Float) -> Boolean): Boolean {\n  for (element in this) if (predicate(element)) return false\n  return
true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n *^npublic inline fun DoubleArray.none(predicate:
(Double) -> Boolean): Boolean {\n  for (element in this) if (predicate(element)) return false\n  return
true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n *^npublic inline fun
BooleanArray.none(predicate: (Boolean) -> Boolean): Boolean {\n  for (element in this) if (predicate(element))
return false\n  return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n *^npublic inline fun CharArray.none(predicate:
(Char) -> Boolean): Boolean {\n  for (element in this) if (predicate(element)) return false\n  return
true\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself afterwards.\n
*\n *^n@SinceKotlin("1.4")\n *^nkotlin.internal.InlineOnly\n *^npublic inline fun <T> Array<out T>.onEach(action: (T) ->
Unit): Array<out T> {\n  return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element and returns the array itself afterwards.\n
*\n *^n@SinceKotlin("1.4")\n *^nkotlin.internal.InlineOnly\n *^npublic inline fun ByteArray.onEach(action: (Byte) ->
Unit): ByteArray {\n  return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element and returns the array itself afterwards.\n
*\n *^n@SinceKotlin("1.4")\n *^nkotlin.internal.InlineOnly\n *^npublic inline fun ShortArray.onEach(action: (Short) ->
Unit): ShortArray {\n  return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element and returns the array itself afterwards.\n
*\n *^n@SinceKotlin("1.4")\n *^nkotlin.internal.InlineOnly\n *^npublic inline fun IntArray.onEach(action: (Int) -> Unit):
IntArray {\n  return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on
each element and returns the array itself afterwards.\n
*\n *^n@SinceKotlin("1.4")\n *^nkotlin.internal.InlineOnly\n *^npublic inline fun LongArray.onEach(action: (Long) ->
Unit): LongArray {\n  return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element and returns the array itself afterwards.\n
*\n *^n@SinceKotlin("1.4")\n *^nkotlin.internal.InlineOnly\n *^npublic inline fun FloatArray.onEach(action: (Float) ->
Unit): FloatArray {\n  return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element and returns the array itself afterwards.\n
*\n *^n@SinceKotlin("1.4")\n *^nkotlin.internal.InlineOnly\n *^npublic inline fun DoubleArray.onEach(action: (Double) ->
Unit): DoubleArray {\n  return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given

```

[action] on each element and returns the array itself afterwards.

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.onEach(action: (Boolean) -> Unit): BooleanArray {\n    return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself afterwards.\n */
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.onEach(action: (Char) -> Unit): CharArray {\n    return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.onEachIndexed(action: (index: Int, T) -> Unit): Array<out T> {\n    return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.onEachIndexed(action: (index: Int, Byte) -> Unit): ByteArray {\n    return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.onEachIndexed(action: (index: Int, Short) -> Unit): ShortArray {\n    return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.onEachIndexed(action: (index: Int, Int) -> Unit): IntArray {\n    return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.onEachIndexed(action: (index: Int, Long) -> Unit): LongArray {\n    return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.onEachIndexed(action: (index: Int, Float) -> Unit): FloatArray {\n    return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.onEachIndexed(action: (index: Int, Double) -> Unit): DoubleArray {\n    return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.onEachIndexed(action: (index: Int, Boolean) -> Unit): BooleanArray {\n    return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.onEachIndexed(action: (index: Int, Char) -> Unit): CharArray {\n    return apply { forEachIndexed(action) }\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its
```

receiver is empty.

```

    @param [operation] function that takes current accumulator value and an element,
    and calculates the next accumulator value.
    @sample samples.collections.Collections.Aggregates.reduce
    *^/public inline fun <S, T : S> Array<out T>.reduce(operation: (acc: S, T) -> S): S {
        if (isEmpty())
            throw UnsupportedOperationException("Empty array can't be reduced.")
        var accumulator: S = this[0]
        for (index in 1..lastIndex) {
            accumulator = operation(accumulator, this[index])
        }
        return accumulator
    }
    * Accumulates value starting with the first element and applying [operation] from left to right
    * to current accumulator value and each element.
    * Throws an exception if this array is empty. If the array can be empty
    in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is empty.
    @param [operation] function that takes current accumulator value and an element,
    and calculates the next accumulator value.
    @sample samples.collections.Collections.Aggregates.reduce
    *^/public inline fun
    ByteArray.reduce(operation: (acc: Byte, Byte) -> Byte): Byte {
        if (isEmpty())
            throw UnsupportedOperationException("Empty array can't be reduced.")
        var accumulator = this[0]
        for (index in 1..lastIndex) {
            accumulator = operation(accumulator, this[index])
        }
        return accumulator
    }
    * Accumulates value starting with the first element and applying [operation] from left to right
    * to current accumulator value and each element.
    * Throws an exception if this array is empty. If the array can be empty
    in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is empty.
    @param [operation] function that takes current accumulator value and an element,
    and calculates the next accumulator value.
    @sample samples.collections.Collections.Aggregates.reduce
    *^/public inline fun
    ShortArray.reduce(operation: (acc: Short, Short) -> Short): Short {
        if (isEmpty())
            throw UnsupportedOperationException("Empty array can't be reduced.")
        var accumulator = this[0]
        for (index in 1..lastIndex) {
            accumulator = operation(accumulator, this[index])
        }
        return accumulator
    }
    * Accumulates value starting with the first element and applying [operation] from left to right
    * to current accumulator value and each element.
    * Throws an exception if this array is empty. If the array can be empty
    in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is empty.
    @param [operation] function that takes current accumulator value and an element,
    and calculates the next accumulator value.
    @sample samples.collections.Collections.Aggregates.reduce
    *^/public inline fun
    IntArray.reduce(operation: (acc: Int, Int) -> Int): Int {
        if (isEmpty())
            throw UnsupportedOperationException("Empty array can't be reduced.")
        var accumulator = this[0]
        for (index in 1..lastIndex) {
            accumulator = operation(accumulator, this[index])
        }
        return accumulator
    }
    * Accumulates value starting with the first element and applying [operation] from left to right
    * to current accumulator value and each element.
    * Throws an exception if this array is empty. If the array can be empty
    in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is empty.
    @param [operation] function that takes current accumulator value and an element,
    and calculates the next accumulator value.
    @sample samples.collections.Collections.Aggregates.reduce
    *^/public inline fun
    LongArray.reduce(operation: (acc: Long, Long) -> Long): Long {
        if (isEmpty())
            throw UnsupportedOperationException("Empty array can't be reduced.")
        var accumulator = this[0]
        for (index in 1..lastIndex) {
            accumulator = operation(accumulator, this[index])
        }
        return accumulator
    }
    * Accumulates value starting with the first element and applying [operation] from left to right
    * to current accumulator value and each element.
    * Throws an exception if this array is empty. If the array can be empty
    in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is empty.
    @param [operation] function that takes current accumulator value and an element,
    and calculates the next accumulator value.
    @sample samples.collections.Collections.Aggregates.reduce
    *^/public inline fun
    FloatArray.reduce(operation: (acc: Float, Float) -> Float): Float {
        if (isEmpty())
            throw UnsupportedOperationException("Empty array can't be reduced.")
        var accumulator = this[0]
        for (index in 1..lastIndex) {
            accumulator = operation(accumulator, this[index])
        }
        return accumulator
    }
    * Accumulates value starting with the first element and applying [operation] from left to right
    * to current accumulator value and each element.
    * Throws an exception if this array is empty. If the array can be empty
    in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is empty.

```

@param [operation] function that takes current accumulator value and an element,
 and calculates the next accumulator value.

```

    @sample samples.collections.Collections.Aggregates.reduce
    public inline fun DoubleArray.reduce(operation: (acc: Double, Double) -> Double): Double {
      if (isEmpty()) throw UnsupportedOperationException("Empty array can't be reduced.")
      var accumulator = this[0]
      for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
      }
      return accumulator
    }
  
```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is empty.

@param [operation] function that takes current accumulator value and an element,
 and calculates the next accumulator value.

```

    @sample samples.collections.Collections.Aggregates.reduce
    public inline fun BooleanArray.reduce(operation: (acc: Boolean, Boolean) -> Boolean): Boolean {
      if (isEmpty()) throw UnsupportedOperationException("Empty array can't be reduced.")
      var accumulator = this[0]
      for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
      }
      return accumulator
    }
  
```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is empty.

@param [operation] function that takes current accumulator value and an element,
 and calculates the next accumulator value.

```

    @sample samples.collections.Collections.Aggregates.reduce
    public inline fun CharArray.reduce(operation: (acc: Char, Char) -> Char): Char {
      if (isEmpty()) throw UnsupportedOperationException("Empty array can't be reduced.")
      var accumulator = this[0]
      for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
      }
      return accumulator
    }
  
```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element with its index in the original array. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.

@param [operation] function that takes the index of an element, current accumulator value and the element itself,
 and calculates the next accumulator value.

```

    @sample samples.collections.Collections.Aggregates.reduce
    public inline fun <S, T : S> Array<out T>.reduceIndexed(operation: (index: Int, acc: S, T) -> S): S {
      if (isEmpty()) throw UnsupportedOperationException("Empty array can't be reduced.")
      var accumulator: S = this[0]
      for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
      }
      return accumulator
    }
  
```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element with its index in the original array. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.

@param [operation] function that takes the index of an element, current accumulator value and the element itself,
 and calculates the next accumulator value.

```

    @sample samples.collections.Collections.Aggregates.reduce
    public inline fun ByteArray.reduceIndexed(operation: (index: Int, acc: Byte, Byte) -> Byte): Byte {
      if (isEmpty()) throw UnsupportedOperationException("Empty array can't be reduced.")
      var accumulator = this[0]
      for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
      }
      return accumulator
    }
  
```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element with its index in the original array. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.

@param [operation] function that takes the index of an element, current accumulator value and the element itself,
 and calculates the next accumulator value.

```

    @sample samples.collections.Collections.Aggregates.reduce
    public inline fun ShortArray.reduceIndexed(operation: (index: Int, acc: Short, Short) -> Short): Short {
      if (isEmpty()) throw UnsupportedOperationException("Empty array can't be reduced.")
      var accumulator = this[0]
      for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
      }
      return accumulator
    }
  
```

accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n * ^\npublic inline fun

```
IntArray.reduceIndexed(operation: (index: Int, acc: Int, Int) -> Int): Int {\n    if (isEmpty())\n        throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(index, accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n * ^\npublic inline fun
```

LongArray.reduceIndexed(operation: (index: Int, acc: Long, Long) -> Long): Long {\n if (isEmpty())\n throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n * ^\npublic inline fun

```
FloatArray.reduceIndexed(operation: (index: Int, acc: Float, Float) -> Float): Float {\n    if (isEmpty())\n        throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(index, accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n * ^\npublic inline fun
```

DoubleArray.reduceIndexed(operation: (index: Int, acc: Double, Double) -> Double): Double {\n if (isEmpty())\n throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n * ^\npublic inline fun

```
BooleanArray.reduceIndexed(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): Boolean {\n    if (isEmpty())\n        throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(index, accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use
```

[reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.

```

n * n * @param [operation] function
that takes the index of an element, current accumulator value and the element itself,
n * and calculates the next
accumulator value.
n * n * @sample samples.collections.Collections.Aggregates.reduce
n * n * public inline fun
CharArray.reduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): Char {
n * if (isEmpty())
n * throw
UnsupportedOperationException("Empty array can't be reduced.")
n * var accumulator = this[0]
n * for (index in
1..lastIndex) {
n * accumulator = operation(index, accumulator, this[index])
n * }
n * return
accumulator
n * }
n * n * Accumulates value starting with the first element and applying [operation] from left to
right
n * to current accumulator value and each element with its index in the original array.
n * n * Returns `null` if
the array is empty.
n * n * @param [operation] function that takes the index of an element, current accumulator
value and the element itself,
n * and calculates the next accumulator value.
n * n * @sample
samples.collections.Collections.Aggregates.reduceOrNull
n * n * @SinceKotlin("1.4")
n * n * public inline fun <S, T : S>
Array<out T>.reduceIndexedOrNull(operation: (index: Int, acc: S, T) -> S): S? {
n * if (isEmpty())
n * return
null
n * var accumulator: S = this[0]
n * for (index in 1..lastIndex) {
n * accumulator = operation(index,
accumulator, this[index])
n * }
n * return accumulator
n * }
n * n * Accumulates value starting with the first
element and applying [operation] from left to right
n * to current accumulator value and each element with its index
in the original array.
n * n * Returns `null` if the array is empty.
n * n * @param [operation] function that takes the
index of an element, current accumulator value and the element itself,
n * and calculates the next accumulator
value.
n * n * @sample samples.collections.Collections.Aggregates.reduceOrNull
n * n * @SinceKotlin("1.4")
n * n * public inline fun ByteArray.reduceIndexedOrNull(operation: (index: Int, acc: Byte,
Byte) -> Byte): Byte? {
n * if (isEmpty())
n * return null
n * var accumulator = this[0]
n * for (index in
1..lastIndex) {
n * accumulator = operation(index, accumulator, this[index])
n * }
n * return
accumulator
n * }
n * n * Accumulates value starting with the first element and applying [operation] from left to
right
n * to current accumulator value and each element with its index in the original array.
n * n * Returns `null` if
the array is empty.
n * n * @param [operation] function that takes the index of an element, current accumulator
value and the element itself,
n * and calculates the next accumulator value.
n * n * @sample
samples.collections.Collections.Aggregates.reduceOrNull
n * n * @SinceKotlin("1.4")
n * n * public inline fun
ShortArray.reduceIndexedOrNull(operation: (index: Int, acc: Short, Short) -> Short): Short? {
n * if (isEmpty())
n * return null
n * var accumulator = this[0]
n * for (index in 1..lastIndex) {
n * accumulator = operation(index,
accumulator, this[index])
n * }
n * return accumulator
n * }
n * n * Accumulates value starting with the first
element and applying [operation] from left to right
n * to current accumulator value and each element with its index
in the original array.
n * n * Returns `null` if the array is empty.
n * n * @param [operation] function that takes the
index of an element, current accumulator value and the element itself,
n * and calculates the next accumulator
value.
n * n * @sample samples.collections.Collections.Aggregates.reduceOrNull
n * n * @SinceKotlin("1.4")
n * n * public inline fun IntArray.reduceIndexedOrNull(operation: (index: Int, acc: Int, Int) ->
Int): Int? {
n * if (isEmpty())
n * return null
n * var accumulator = this[0]
n * for (index in 1..lastIndex) {
n * accumulator = operation(index, accumulator, this[index])
n * }
n * return accumulator
n * }
n * n * Accumulates
value starting with the first element and applying [operation] from left to right
n * to current accumulator value and
each element with its index in the original array.
n * n * Returns `null` if the array is empty.
n * n * @param
[operation] function that takes the index of an element, current accumulator value and the element itself,
n * and
calculates the next accumulator value.
n * n * @sample samples.collections.Collections.Aggregates.reduceOrNull
n * n * @SinceKotlin("1.4")
n * n * public inline fun LongArray.reduceIndexedOrNull(operation: (index: Int, acc: Long,
Long) -> Long): Long? {
n * if (isEmpty())
n * return null
n * var accumulator = this[0]
n * for (index in
1..lastIndex) {
n * accumulator = operation(index, accumulator, this[index])
n * }
n * return
accumulator
n * }
n * n * Accumulates value starting with the first element and applying [operation] from left to
right
n * to current accumulator value and each element with its index in the original array.
n * n * Returns `null` if
the array is empty.
n * n * @param [operation] function that takes the index of an element, current accumulator
value and the element itself,
n * and calculates the next accumulator value.
n * n * @sample
samples.collections.Collections.Aggregates.reduceOrNull
n * n * @SinceKotlin("1.4")
n * n * public inline fun

```



```

FloatArray.reduceIndexedOrNull(operation: (index: Int, acc: Float, Float) -> Float): Float? {\n  if (isEmpty())\n  return null\n  var accumulator = this[0]\n  for (index in 1..lastIndex) {\n    accumulator = operation(index,\n  accumulator, this[index])\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the first\n  element and applying [operation] from left to right\n * to current accumulator value and each element with its index\n  in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the\n  index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator\n  value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n\n*/\n@SinceKotlin("1.4")\npublic inline fun DoubleArray.reduceIndexedOrNull(operation: (index: Int, acc:\n  Double, Double) -> Double): Double? {\n  if (isEmpty())\n    return null\n  var accumulator = this[0]\n  for\n  (index in 1..lastIndex) {\n    accumulator = operation(index, accumulator, this[index])\n  }\n  return\n  accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to\n  right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator\n  value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample\n  samples.collections.Collections.Aggregates.reduceOrNull\n\n*/\n@SinceKotlin("1.4")\npublic inline fun\n  BooleanArray.reduceIndexedOrNull(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): Boolean? {\n  if\n  (isEmpty())\n    return null\n  var accumulator = this[0]\n  for (index in 1..lastIndex) {\n    accumulator =\n  operation(index, accumulator, this[index])\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting\n  with the first element and applying [operation] from left to right\n * to current accumulator value and each element\n  with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation]\n  function that takes the index of an element, current accumulator value and the element\n  itself,\n * and calculates the next accumulator value.\n * \n * @sample\n  samples.collections.Collections.Aggregates.reduceOrNull\n\n*/\n@SinceKotlin("1.4")\npublic inline fun\n  CharArray.reduceIndexedOrNull(operation: (index: Int, acc: Char, Char) -> Char): Char? {\n  if\n  (isEmpty())\n    return null\n  var accumulator = this[0]\n  for (index in\n  1..lastIndex) {\n    accumulator = operation(index, accumulator, this[index])\n  }\n  return\n  accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to\n  right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next\n  accumulator value.\n * \n * @sample\n  samples.collections.Collections.Aggregates.reduceOrNull\n\n*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S>\n  Array<out T>.reduceOrNull(operation: (acc: S, T) -> S): S? {\n  if (isEmpty())\n    return null\n  var\n  accumulator: S = this[0]\n  for (index in 1..lastIndex) {\n    accumulator = operation(accumulator, this[index])\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation]\n  from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next\n  accumulator value.\n * \n * @sample\n  samples.collections.Collections.Aggregates.reduceOrNull\n\n*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun\n  ByteArray.reduceOrNull(operation: (acc: Byte, Byte) -> Byte): Byte? {\n  if (isEmpty())\n    return null\n  var\n  accumulator = this[0]\n  for (index in 1..lastIndex) {\n    accumulator = operation(accumulator, this[index])\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation]\n  from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next\n  accumulator value.\n * \n * @sample\n  samples.collections.Collections.Aggregates.reduceOrNull\n\n*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun\n  ShortArray.reduceOrNull(operation: (acc: Short, Short) -> Short): Short? {\n  if (isEmpty())\n    return null\n  var\n  accumulator = this[0]\n  for (index in 1..lastIndex) {\n    accumulator = operation(accumulator, this[index])\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation]\n  from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * \n *

```

```

\n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
IntArray.reduceOrNull(operation: (acc: Int, Int) -> Int): Int? {\n if (isEmpty())\n return null\n var
accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n
}\n return accumulator}\n\n/**\n * Accumulates value starting with the first element and applying [operation]
from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n *
\n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
LongArray.reduceOrNull(operation: (acc: Long, Long) -> Long): Long? {\n if (isEmpty())\n return null\n
var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n
}\n return accumulator}\n\n/**\n * Accumulates value starting with the first element and applying [operation]
from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n *
\n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
FloatArray.reduceOrNull(operation: (acc: Float, Float) -> Float): Float? {\n if (isEmpty())\n return null\n var
accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n
}\n return accumulator}\n\n/**\n * Accumulates value starting with the first element and applying [operation]
from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n *
\n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
DoubleArray.reduceOrNull(operation: (acc: Double, Double) -> Double): Double? {\n if (isEmpty())\n return
null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator,
this[index])\n }\n return accumulator}\n\n/**\n * Accumulates value starting with the first element and
applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if
the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n *
and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
BooleanArray.reduceOrNull(operation: (acc: Boolean, Boolean) -> Boolean): Boolean? {\n if (isEmpty())\n
return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator =
operation(accumulator, this[index])\n }\n return accumulator}\n\n/**\n * Accumulates value starting with the
first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n *
Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and
an element,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
CharArray.reduceOrNull(operation: (acc: Char, Char) -> Char): Char? {\n if (isEmpty())\n return null\n var
accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n
}\n return accumulator}\n\n/**\n * Accumulates value starting with the last element and applying [operation]
from right to left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is
empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null`
when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator
value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n * \n@npublic inline fun <S, T : S> Array<out

```

```

T>.reduceRight(operation: (T, acc: S) -> S): S {\n  var index = lastIndex\n  if (index < 0) throw
UnsupportedOperationException("Empty array can't be reduced.")\n  var accumulator: S = get(index--)\n  while
(index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and
current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an
expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n *
@param [operation] function that takes an element and current accumulator value,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline
fun ByteArray.reduceRight(operation: (Byte, acc: Byte) -> Byte): Byte {\n  var index = lastIndex\n  if (index < 0)
throw UnsupportedOperationException("Empty array can't be reduced.")\n  var accumulator = get(index--)\n  while
(index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n  }\n  return
accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the
array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its
receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n *
and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun ShortArray.reduceRight(operation:
(Short, acc: Short) -> Short): Short {\n  var index = lastIndex\n  if (index < 0) throw
UnsupportedOperationException("Empty array can't be reduced.")\n  var accumulator = get(index--)\n  while
(index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and
current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an
expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n *
@param [operation] function that takes an element and current accumulator value,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline
fun IntArray.reduceRight(operation: (Int, acc: Int) -> Int): Int {\n  var index = lastIndex\n  if (index < 0) throw
UnsupportedOperationException("Empty array can't be reduced.")\n  var accumulator = get(index--)\n  while
(index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and
current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an
expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n *
@param [operation] function that takes an element and current accumulator value,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline
fun LongArray.reduceRight(operation: (Long, acc: Long) -> Long): Long {\n  var index = lastIndex\n  if (index <
0) throw UnsupportedOperationException("Empty array can't be reduced.")\n  var accumulator = get(index--)\n  while
(index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n  }\n  return
accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the
array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its
receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n *
and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun FloatArray.reduceRight(operation:
(Float, acc: Float) -> Float): Float {\n  var index = lastIndex\n  if (index < 0) throw
UnsupportedOperationException("Empty array can't be reduced.")\n  var accumulator = get(index--)\n  while
(index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and
current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an
expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n *
@param [operation] function that takes an element and current accumulator value,\n * and calculates the next
accumulator value.\n * \n * @sample

```

@param [operation] function that takes an element and current accumulator value, \n * and calculates the next accumulator value. \n * \n * @sample samples.collections.Collections.Aggregates.reduceRight \n */ \n public inline fun DoubleArray.reduceRight(operation: (Double, acc: Double) -> Double): Double { \n var index = lastIndex \n if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.") \n var accumulator = get(index--) \n while (index >= 0) { \n accumulator = operation(get(index--), accumulator) \n } \n return accumulator \n } \n \n /** \n * Accumulates value starting with the last element and applying [operation] from right to left \n * to each element and current accumulator value. \n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way, \n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty. \n * \n * @param [operation] function that takes an element and current accumulator value, \n * and calculates the next accumulator value. \n * \n * @sample samples.collections.Collections.Aggregates.reduceRight \n */ \n public inline fun BooleanArray.reduceRight(operation: (Boolean, acc: Boolean) -> Boolean): Boolean { \n var index = lastIndex \n if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.") \n var accumulator = get(index--) \n while (index >= 0) { \n accumulator = operation(get(index--), accumulator) \n } \n return accumulator \n } \n \n /** \n * Accumulates value starting with the last element and applying [operation] from right to left \n * to each element and current accumulator value. \n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way, \n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty. \n * \n * @param [operation] function that takes an element and current accumulator value, \n * and calculates the next accumulator value. \n * \n * @sample samples.collections.Collections.Aggregates.reduceRight \n */ \n public inline fun CharArray.reduceRight(operation: (Char, acc: Char) -> Char): Char { \n var index = lastIndex \n if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.") \n var accumulator = get(index--) \n while (index >= 0) { \n accumulator = operation(get(index--), accumulator) \n } \n return accumulator \n } \n \n /** \n * Accumulates value starting with the last element and applying [operation] from right to left \n * to each element with its index in the original array and current accumulator value. \n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way, \n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty. \n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value, \n * and calculates the next accumulator value. \n * \n * @sample samples.collections.Collections.Aggregates.reduceRight \n */ \n public inline fun <S, T : S> Array<out T>.reduceRightIndexed(operation: (index: Int, T, acc: S) -> S): S { \n var index = lastIndex \n if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.") \n var accumulator: S = get(index--) \n while (index >= 0) { \n accumulator = operation(index, get(index), accumulator) \n --index \n } \n return accumulator \n } \n \n /** \n * Accumulates value starting with the last element and applying [operation] from right to left \n * to each element with its index in the original array and current accumulator value. \n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way, \n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty. \n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value, \n * and calculates the next accumulator value. \n * \n * @sample samples.collections.Collections.Aggregates.reduceRight \n */ \n public inline fun ByteArray.reduceRightIndexed(operation: (index: Int, Byte, acc: Byte) -> Byte): Byte { \n var index = lastIndex \n if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.") \n var accumulator = get(index--) \n while (index >= 0) { \n accumulator = operation(index, get(index), accumulator) \n --index \n } \n return accumulator \n } \n \n /** \n * Accumulates value starting with the last element and applying [operation] from right to left \n * to each element with its index in the original array and current accumulator value. \n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way, \n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty. \n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value, \n * and calculates the next accumulator value. \n * \n * @sample samples.collections.Collections.Aggregates.reduceRight \n */ \n public inline fun

```

ShortArray.reduceRightIndexed(operation: (index: Int, Short, acc: Short) -> Short): Short {
  var index = lastIndex
  if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
  var accumulator = get(index--)
  while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
    --index
  }
  return accumulator
}

```

* Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.
 * Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.
 @sample
 samples.collections.Collections.Aggregates.reduceRight

```

IntArray.reduceRightIndexed(operation: (index: Int, Int, acc: Int) -> Int): Int {
  var index = lastIndex
  if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
  var accumulator = get(index--)
  while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
    --index
  }
  return accumulator
}

```

* Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.
 * Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.
 @sample
 samples.collections.Collections.Aggregates.reduceRight

```

LongArray.reduceRightIndexed(operation: (index: Int, Long, acc: Long) -> Long): Long {
  var index = lastIndex
  if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
  var accumulator = get(index--)
  while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
    --index
  }
  return accumulator
}

```

* Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.
 * Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.
 @sample
 samples.collections.Collections.Aggregates.reduceRight

```

FloatArray.reduceRightIndexed(operation: (index: Int, Float, acc: Float) -> Float): Float {
  var index = lastIndex
  if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
  var accumulator = get(index--)
  while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
    --index
  }
  return accumulator
}

```

* Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.
 * Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.
 @sample
 samples.collections.Collections.Aggregates.reduceRight

```

DoubleArray.reduceRightIndexed(operation: (index: Int, Double, acc: Double) -> Double): Double {
  var index = lastIndex
  if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
  var accumulator = get(index--)
  while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
    --index
  }
  return accumulator
}

```

* Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.
 * Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.
 @sample

```

samples.collections.Collections.Aggregates.reduceRight\n *^\npublic inline fun
BooleanArray.reduceRightIndexed(operation: (index: Int, Boolean, acc: Boolean) -> Boolean): Boolean {\n  var
index = lastIndex\n  if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n
var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator = operation(index, get(index),
accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last
element and applying [operation] from right to left\n * to each element with its index in the original array and
current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an
expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator
value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n *^\npublic inline fun
CharArray.reduceRightIndexed(operation: (index: Int, Char, acc: Char) -> Char): Char {\n  var index = lastIndex\n
if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n  var accumulator =
get(index--)\n  while (index >= 0) {\n    accumulator = operation(index, get(index), accumulator)\n    --index\n
}\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation]
from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n *
Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the
element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n *^\n@SinceKotlin("1.4")\npublic inline fun <S,
T : S> Array<out T>.reduceRightIndexedOrNull(operation: (index: Int, T, acc: S) -> S): S? {\n  var index =
lastIndex\n  if (index < 0) return null\n  var accumulator: S = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n *
Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with
its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator value,\n *
and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n *^\n@SinceKotlin("1.4")\npublic inline fun
ByteArray.reduceRightIndexedOrNull(operation: (index: Int, Byte, acc: Byte) -> Byte): Byte? {\n  var index =
lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n *
Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with
its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator value,\n *
and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n *^\n@SinceKotlin("1.4")\npublic inline fun
ShortArray.reduceRightIndexedOrNull(operation: (index: Int, Short, acc: Short) -> Short): Short? {\n  var index =
lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n *
Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with
its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator value,\n *
and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n *^\n@SinceKotlin("1.4")\npublic inline fun
IntArray.reduceRightIndexedOrNull(operation: (index: Int, Int, acc: Int) -> Int): Int? {\n  var index = lastIndex\n
if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator =
operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n * Accumulates
value starting with the last element and applying [operation] from right to left\n * to each element with its index in
the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param

```

[operation] function that takes the index of an element, the element itself and current accumulator value, \n * and calculates the next accumulator value. \n * \n * @sample

```

samples.collections.Collections.Aggregates.reduceRightOrNull \n * \n @SinceKotlin("1.4") \n public inline fun
LongArray.reduceRightIndexedOrNull(operation: (index: Int, Long, acc: Long) -> Long): Long? { \n   var index =
lastIndex \n   if (index < 0) return null \n   var accumulator = get(index--) \n   while (index >= 0) { \n
accumulator = operation(index, get(index), accumulator) \n   --index \n   } \n   return accumulator \n } \n \n /** \n *
Accumulates value starting with the last element and applying [operation] from right to left \n * to each element with
its index in the original array and current accumulator value. \n * \n * Returns `null` if the array is empty. \n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator value, \n
* and calculates the next accumulator value. \n * \n * @sample

```

```

samples.collections.Collections.Aggregates.reduceRightOrNull \n * \n @SinceKotlin("1.4") \n public inline fun
FloatArray.reduceRightIndexedOrNull(operation: (index: Int, Float, acc: Float) -> Float): Float? { \n   var index =
lastIndex \n   if (index < 0) return null \n   var accumulator = get(index--) \n   while (index >= 0) { \n
accumulator = operation(index, get(index), accumulator) \n   --index \n   } \n   return accumulator \n } \n \n /** \n *
Accumulates value starting with the last element and applying [operation] from right to left \n * to each element with
its index in the original array and current accumulator value. \n * \n * Returns `null` if the array is empty. \n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator value, \n
* and calculates the next accumulator value. \n * \n * @sample

```

```

samples.collections.Collections.Aggregates.reduceRightOrNull \n * \n @SinceKotlin("1.4") \n public inline fun
DoubleArray.reduceRightIndexedOrNull(operation: (index: Int, Double, acc: Double) -> Double): Double? { \n   var
index = lastIndex \n   if (index < 0) return null \n   var accumulator = get(index--) \n   while (index >= 0) { \n
accumulator = operation(index, get(index), accumulator) \n   --index \n   } \n   return accumulator \n } \n \n /** \n *
Accumulates value starting with the last element and applying [operation] from right to left \n * to each element with
its index in the original array and current accumulator value. \n * \n * Returns `null` if the array is empty. \n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator value, \n
* and calculates the next accumulator value. \n * \n * @sample

```

```

samples.collections.Collections.Aggregates.reduceRightOrNull \n * \n @SinceKotlin("1.4") \n public inline fun
BooleanArray.reduceRightIndexedOrNull(operation: (index: Int, Boolean, acc: Boolean) -> Boolean): Boolean? { \n   \n
var index = lastIndex \n   if (index < 0) return null \n   var accumulator = get(index--) \n   while (index >= 0) { \n
accumulator = operation(index, get(index), accumulator) \n   --index \n   } \n   return accumulator \n } \n \n /** \n *
Accumulates value starting with the last element and applying [operation] from right to left \n * to each element with
its index in the original array and current accumulator value. \n * \n * Returns `null` if the array is empty. \n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator value, \n
* and calculates the next accumulator value. \n * \n * @sample

```

```

samples.collections.Collections.Aggregates.reduceRightOrNull \n * \n @SinceKotlin("1.4") \n public inline fun
CharArray.reduceRightIndexedOrNull(operation: (index: Int, Char, acc: Char) -> Char): Char? { \n   var index =
lastIndex \n   if (index < 0) return null \n   var accumulator = get(index--) \n   while (index >= 0) { \n
accumulator = operation(index, get(index), accumulator) \n   --index \n   } \n   return accumulator \n } \n \n /** \n *
Accumulates value starting with the last element and applying [operation] from right to left \n * to each element and
current accumulator value. \n * \n * Returns `null` if the array is empty. \n * \n * @param [operation] function that
takes an element and current accumulator value, \n * and calculates the next accumulator value. \n * \n * @sample

```

```

samples.collections.Collections.Aggregates.reduceRightOrNull \n
* \n @SinceKotlin("1.4") \n @WasExperimental(ExperimentalStdlibApi::class) \n public inline fun <S, T : S>
Array<out T>.reduceRightOrNull(operation: (T, acc: S) -> S): S? { \n   var index = lastIndex \n   if (index < 0)
return null \n   var accumulator: S = get(index--) \n   while (index >= 0) { \n   accumulator = operation(get(index--
), accumulator) \n   } \n   return accumulator \n } \n \n /** \n * Accumulates value starting with the last element and
applying [operation] from right to left \n * to each element and current accumulator value. \n * \n * Returns `null` if
the array is empty. \n * \n * @param [operation] function that takes an element and current accumulator value, \n *

```

and calculates the next accumulator value.\n * \n * @sample

```

samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
ByteArray.reduceRightOrNull(operation: (Byte, acc: Byte) -> Byte): Byte? {\n  var index = lastIndex\n  if (index
< 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator =
operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the
last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n *
Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current
accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
ShortArray.reduceRightOrNull(operation: (Short, acc: Short) -> Short): Short? {\n  var index = lastIndex\n  if
(index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator =
operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the
last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n *
Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current
accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
IntArray.reduceRightOrNull(operation: (Int, acc: Int) -> Int): Int? {\n  var index = lastIndex\n  if (index < 0)
return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator = operation(get(index--),
accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and
applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns `null` if
the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n *
and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
LongArray.reduceRightOrNull(operation: (Long, acc: Long) -> Long): Long? {\n  var index = lastIndex\n  if
(index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator =
operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the
last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n *
Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current
accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
FloatArray.reduceRightOrNull(operation: (Float, acc: Float) -> Float): Float? {\n  var index = lastIndex\n  if
(index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator =
operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the
last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n *
Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current
accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
DoubleArray.reduceRightOrNull(operation: (Double, acc: Double) -> Double): Double? {\n  var index =
lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value
starting with the last element and applying [operation] from right to left\n * to each element and current accumulator
value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and

```


current accumulator value, and calculates the next accumulator value. @sample

```

samples.collections.Collections.Aggregates.reduceRightOrNull
*/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)npublic inline fun
BooleanArray.reduceRightOrNull(operation: (Boolean, acc: Boolean) -> Boolean): Boolean? {n var index =
lastIndexn if (index < 0) return nulln var accumulator = get(index--)\n while (index >= 0) {n
accumulator = operation(get(index--), accumulator)\n }n return accumulator\n}\n/n/**n * Accumulates value
starting with the last element and applying [operation] from right to leftn * to each element and current accumulator
value.n * n * Returns `null` if the array is empty.n * n * @param [operation] function that takes an element and
current accumulator value, and calculates the next accumulator value.n * n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull
*/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)npublic inline fun
CharArray.reduceRightOrNull(operation: (Char, acc: Char) -> Char): Char? {n var index = lastIndexn if (index
< 0) return nulln var accumulator = get(index--)\n while (index >= 0) {n accumulator =
operation(get(index--), accumulator)\n }n return accumulator\n}\n/n/**n * Returns a list containing successive
accumulation values generated by applying [operation] from left to rightn * to each element and current
accumulator value that starts with [initial] value.n * n * Note that `acc` value passed to [operation] function should
not be mutated;n * otherwise it would affect the previous value in resulting list.n * n * @param [operation]
function that takes current accumulator value and an element, and calculates the next accumulator value.n * n *
@sample samples.collections.Collections.Aggregates.runningFold
*/n@SinceKotlin("1.4")npublic inline fun
<T, R> Array<out T>.runningFold(initial: R, operation: (acc: R, T) -> R): List<R> {n if (isEmpty()) return
listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for
(element in this) {n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }n
return result\n}\n/n/**n * Returns a list containing successive accumulation values generated by applying
[operation] from left to rightn * to each element and current accumulator value that starts with [initial] value.n *
n * Note that `acc` value passed to [operation] function should not be mutated;n * otherwise it would affect the
previous value in resulting list.n * n * @param [operation] function that takes current accumulator value and an
element, and calculates the next accumulator value.n * n * @sample
samples.collections.Collections.Aggregates.runningFold
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun <R> ByteArray.runningFold(initial: R,
operation: (acc: R, Byte) -> R): List<R> {n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size
+ 1).apply { add(initial) }\n var accumulator = initial\n for (element in this) {n accumulator =
operation(accumulator, element)\n result.add(accumulator)\n }n return result\n}\n/n/**n * Returns a list
containing successive accumulation values generated by applying [operation] from left to rightn * to each element
and current accumulator value that starts with [initial] value.n * n * Note that `acc` value passed to [operation]
function should not be mutated;n * otherwise it would affect the previous value in resulting list.n * n * @param
[operation] function that takes current accumulator value and an element, and calculates the next accumulator
value.n * n * @sample samples.collections.Collections.Aggregates.runningFold
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun <R> ShortArray.runningFold(initial: R,
operation: (acc: R, Short) -> R): List<R> {n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size
+ 1).apply { add(initial) }\n var accumulator = initial\n for (element in this) {n accumulator =
operation(accumulator, element)\n result.add(accumulator)\n }n return result\n}\n/n/**n * Returns a list
containing successive accumulation values generated by applying [operation] from left to rightn * to each element
and current accumulator value that starts with [initial] value.n * n * Note that `acc` value passed to [operation]
function should not be mutated;n * otherwise it would affect the previous value in resulting list.n * n * @param
[operation] function that takes current accumulator value and an element, and calculates the next accumulator
value.n * n * @sample samples.collections.Collections.Aggregates.runningFold
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun <R> IntArray.runningFold(initial: R,
operation: (acc: R, Int) -> R): List<R> {n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size +

```

```

1).apply { add(initial) }
var accumulator = initial
for (element in this) {
    accumulator =
operation(accumulator, element)
result.add(accumulator)
}
return result
}

/**
 * Returns a list
containing successive accumulation values generated by applying [operation] from left to right
 * to each element
and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation]
function should not be mutated;
 * otherwise it would affect the previous value in resulting list.
 * @param
[operation] function that takes current accumulator value and an element, and calculates the next accumulator
value.
 * @sample samples.collections.Collections.Aggregates.runningFold

*/
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <R> LongArray.runningFold(initial: R,
operation: (acc: R, Long) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size
+ 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator =
operation(accumulator, element)
result.add(accumulator)
    }
    return result
}

/**
 * Returns a list
containing successive accumulation values generated by applying [operation] from left to right
 * to each element
and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation]
function should not be mutated;
 * otherwise it would affect the previous value in resulting list.
 * @param
[operation] function that takes current accumulator value and an element, and calculates the next accumulator
value.
 * @sample samples.collections.Collections.Aggregates.runningFold

*/
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <R> FloatArray.runningFold(initial: R,
operation: (acc: R, Float) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size
+ 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator =
operation(accumulator, element)
result.add(accumulator)
    }
    return result
}

/**
 * Returns a list
containing successive accumulation values generated by applying [operation] from left to right
 * to each element
and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation]
function should not be mutated;
 * otherwise it would affect the previous value in resulting list.
 * @param
[operation] function that takes current accumulator value and an element, and calculates the next accumulator
value.
 * @sample samples.collections.Collections.Aggregates.runningFold

*/
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <R> DoubleArray.runningFold(initial: R,
operation: (acc: R, Double) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result =
ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to
[operation] function should not be mutated;
 * otherwise it would affect the previous value in resulting list.
 * @param
[operation] function that takes current accumulator value and an element, and calculates the next
accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningFold

*/
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <R> BooleanArray.runningFold(initial:
R, operation: (acc: R, Boolean) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result =
ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to
[operation] function should not be mutated;
 * otherwise it would affect the previous value in resulting list.
 * @param
[operation] function that takes current accumulator value and an element, and calculates the next
accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningFold

*/
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <R> CharArray.runningFold(initial: R,
operation: (acc: R, Char) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size
+ 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator =
operation(accumulator, element)
result.add(accumulator)
    }
    return result
}

/**
 * Returns a list
containing successive accumulation values generated by applying [operation] from left to right
 * to each element,

```

its index in the original array and current accumulator value that starts with [initial] value.

Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

@sample

```

samples.collections.Collections.Aggregates.runningFold
*/
@SinceKotlin("1.4")
public inline fun <T, R>
Array<out T>.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {
    if (isEmpty())
        return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value.

Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

@sample

```

samples.collections.Collections.Aggregates.runningFold
*/
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <R>
ByteArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Byte) -> R): List<R> {
    if (isEmpty())
        return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value.

Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

@sample

```

samples.collections.Collections.Aggregates.runningFold
*/
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <R>
ShortArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): List<R> {
    if (isEmpty())
        return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value.

Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

@sample

```

samples.collections.Collections.Aggregates.runningFold
*/
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <R>
IntArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): List<R> {
    if (isEmpty())
        return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value.

Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

@sample

```

samples.collections.Collections.Aggregates.runningFold
*/
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <R>
LongArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Long) -> R): List<R> {
    if (isEmpty())
        return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])

```

```

result.add(accumulator)\n    }\n    return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function
should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R>
FloatArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Float) -> R): List<R> {\n    if (isEmpty())
return listOf(initial)\n    val result = ArrayList<R>(size + 1).apply { add(initial) }\n    var accumulator = initial\n
for (index in indices) {\n        accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n    }\n    return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function
should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R>
DoubleArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Double) -> R): List<R> {\n    if
(isEmpty()) return listOf(initial)\n    val result = ArrayList<R>(size + 1).apply { add(initial) }\n    var accumulator =
initial\n    for (index in indices) {\n        accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n    }\n    return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function
should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R>
BooleanArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Boolean) -> R): List<R> {\n    if
(isEmpty()) return listOf(initial)\n    val result = ArrayList<R>(size + 1).apply { add(initial) }\n    var accumulator =
initial\n    for (index in indices) {\n        accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n    }\n    return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function
should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R>
CharArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): List<R> {\n    if (isEmpty())
return listOf(initial)\n    val result = ArrayList<R>(size + 1).apply { add(initial) }\n    var accumulator = initial\n
for (index in indices) {\n        accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n    }\n    return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element and current accumulator value that
starts with the first element of this array.\n * \n * Note that `acc` value passed to [operation] function should not be
mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that
takes current accumulator value and the element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S>
Array<out T>.runningReduce(operation: (acc: S, T) -> S): List<S> {\n    if (isEmpty()) return emptyList()\n    var

```

```

accumulator: S = this[0]\n  val result = ArrayList<S>(size).apply { add(accumulator) }\n  for (index in 1 until
size) {\n    accumulator = operation(accumulator, this[index])\n    result.add(accumulator)\n  }\n  return
result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from
left to right\n * to each element and current accumulator value that starts with the first element of this array.\n * \n *
@param [operation] function that takes current accumulator value and an element, and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n
*/\n\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.runningReduce(operation:
(acc: Byte, Byte) -> Byte): List<Byte> {\n  if (isEmpty()) return emptyList()\n  var accumulator = this[0]\n  val
result = ArrayList<Byte>(size).apply { add(accumulator) }\n  for (index in 1 until size) {\n    accumulator =
operation(accumulator, this[index])\n    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element
and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function
that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
*/\n\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.runningReduce(operation:
(acc: Short, Short) -> Short): List<Short> {\n  if (isEmpty()) return emptyList()\n  var accumulator = this[0]\n
val result = ArrayList<Short>(size).apply { add(accumulator) }\n  for (index in 1 until size) {\n    accumulator =
operation(accumulator, this[index])\n    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element
and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function
that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
*/\n\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.runningReduce(operation: (acc:
Int, Int) -> Int): List<Int> {\n  if (isEmpty()) return emptyList()\n  var accumulator = this[0]\n  val result =
ArrayList<Int>(size).apply { add(accumulator) }\n  for (index in 1 until size) {\n    accumulator =
operation(accumulator, this[index])\n    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element
and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function
that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
*/\n\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.runningReduce(operation:
(acc: Long, Long) -> Long): List<Long> {\n  if (isEmpty()) return emptyList()\n  var accumulator = this[0]\n
val result = ArrayList<Long>(size).apply { add(accumulator) }\n  for (index in 1 until size) {\n    accumulator =
operation(accumulator, this[index])\n    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element
and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function
that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
*/\n\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.runningReduce(operation:
(acc: Float, Float) -> Float): List<Float> {\n  if (isEmpty()) return emptyList()\n  var accumulator = this[0]\n  val
result = ArrayList<Float>(size).apply { add(accumulator) }\n  for (index in 1 until size) {\n    accumulator =
operation(accumulator, this[index])\n    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element
and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function
that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
*/\n\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.runningReduce(operation:
(acc: Double, Double) -> Double): List<Double> {\n  if (isEmpty()) return emptyList()\n  var accumulator =

```

```

this[0])\n    val result = ArrayList<Double>(size).apply { add(accumulator) }\n    for (index in 1 until size) {\n
accumulator = operation(accumulator, this[index])\n    result.add(accumulator)\n    }\n    return result\n}\n\n/**\n
* Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to
each element and current accumulator value that starts with the first element of this array.\n * \n * @param
[operation] function that takes current accumulator value and an element, and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n
*/\n\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.runningReduce(operation:
(acc: Boolean, Boolean) -> Boolean): List<Boolean> {\n    if (isEmpty()) return emptyList()\n    var accumulator =
this[0]\n    val result = ArrayList<Boolean>(size).apply { add(accumulator) }\n    for (index in 1 until size) {\n
accumulator = operation(accumulator, this[index])\n    result.add(accumulator)\n    }\n    return result\n}\n\n/**\n
* Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to
each element and current accumulator value that starts with the first element of this array.\n * \n * @param
[operation] function that takes current accumulator value and an element, and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n
*/\n\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.runningReduce(operation:
(acc: Char, Char) -> Char): List<Char> {\n    if (isEmpty()) return emptyList()\n    var accumulator = this[0]\n    val
result = ArrayList<Char>(size).apply { add(accumulator) }\n    for (index in 1 until size) {\n    accumulator =
operation(accumulator, this[index])\n    result.add(accumulator)\n    }\n    return result\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element,
its index in the original array and current accumulator value that starts with the first element of this array.\n * \n *
Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the
previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current
accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
*/\n\n@SinceKotlin("1.4")\npublic inline fun <S, T :
S> Array<out T>.runningReduceIndexed(operation: (index: Int, acc: S, T) -> S): List<S> {\n    if (isEmpty()) return
emptyList()\n    var accumulator: S = this[0]\n    val result = ArrayList<S>(size).apply { add(accumulator) }\n    for
(index in 1 until size) {\n    accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n    }\n    return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function that
takes the index of an element, current accumulator value\n * and the element itself, and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n
*/\n\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
ByteArray.runningReduceIndexed(operation: (index: Int, acc: Byte, Byte) -> Byte): List<Byte> {\n    if (isEmpty())
return emptyList()\n    var accumulator = this[0]\n    val result = ArrayList<Byte>(size).apply { add(accumulator)
}\n    for (index in 1 until size) {\n    accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n    }\n    return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function that
takes the index of an element, current accumulator value\n * and the element itself, and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n
*/\n\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
ShortArray.runningReduceIndexed(operation: (index: Int, acc: Short, Short) -> Short): List<Short> {\n    if
(isEmpty()) return emptyList()\n    var accumulator = this[0]\n    val result = ArrayList<Short>(size).apply {
add(accumulator) }\n    for (index in 1 until size) {\n    accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n    }\n    return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function that

```

takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

```

@sample samples.collections.Collections.Aggregates.runningReduce
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun
IntArray.runningReduceIndexed(operation: (index: Int, acc: Int, Int) -> Int): List<Int> {n if (isEmpty()) return
emptyList()n var accumulator = this[0]n val result = ArrayList<Int>(size).apply { add(accumulator) }n for
(index in 1 until size) {n accumulator = operation(index, accumulator, this[index])n
result.add(accumulator)n }n return result}n/n/**n * Returns a list containing successive accumulation
values generated by applying [operation] from left to rightn * to each element, its index in the original array and
current accumulator value that starts with the first element of this array.n * n * @param [operation] function that
takes the index of an element, current accumulator value and the element itself, and calculates the next
accumulator value.n * n * @sample samples.collections.Collections.Aggregates.runningReduce
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun
LongArray.runningReduceIndexed(operation: (index: Int, acc: Long, Long) -> Long): List<Long> {n if
(isEmpty()) return emptyList()n var accumulator = this[0]n val result = ArrayList<Long>(size).apply {
add(accumulator) }n for (index in 1 until size) {n accumulator = operation(index, accumulator, this[index])n
result.add(accumulator)n }n return result}n/n/**n * Returns a list containing successive accumulation
values generated by applying [operation] from left to rightn * to each element, its index in the original array and
current accumulator value that starts with the first element of this array.n * n * @param [operation] function that
takes the index of an element, current accumulator value and the element itself, and calculates the next
accumulator value.n * n * @sample samples.collections.Collections.Aggregates.runningReduce
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun
FloatArray.runningReduceIndexed(operation: (index: Int, acc: Float, Float) -> Float): List<Float> {n if
(isEmpty()) return emptyList()n var accumulator = this[0]n val result = ArrayList<Float>(size).apply {
add(accumulator) }n for (index in 1 until size) {n accumulator = operation(index, accumulator, this[index])n
result.add(accumulator)n }n return result}n/n/**n * Returns a list containing successive accumulation
values generated by applying [operation] from left to rightn * to each element, its index in the original array and
current accumulator value that starts with the first element of this array.n * n * @param [operation] function that
takes the index of an element, current accumulator value and the element itself, and calculates the next
accumulator value.n * n * @sample samples.collections.Collections.Aggregates.runningReduce
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun
DoubleArray.runningReduceIndexed(operation: (index: Int, acc: Double, Double) -> Double): List<Double> {n if
(isEmpty()) return emptyList()n var accumulator = this[0]n val result = ArrayList<Double>(size).apply {
add(accumulator) }n for (index in 1 until size) {n accumulator = operation(index, accumulator, this[index])n
result.add(accumulator)n }n return result}n/n/**n * Returns a list containing successive accumulation
values generated by applying [operation] from left to rightn * to each element, its index in the original array and
current accumulator value that starts with the first element of this array.n * n * @param [operation] function that
takes the index of an element, current accumulator value and the element itself, and calculates the next
accumulator value.n * n * @sample samples.collections.Collections.Aggregates.runningReduce
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun
BooleanArray.runningReduceIndexed(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): List<Boolean>
{n if (isEmpty()) return emptyList()n var accumulator = this[0]n val result =
ArrayList<Boolean>(size).apply { add(accumulator) }n for (index in 1 until size) {n accumulator =
operation(index, accumulator, this[index])n result.add(accumulator)n }n return result}n/n/**n *
Returns a list containing successive accumulation values generated by applying [operation] from left to rightn * to
each element, its index in the original array and current accumulator value that starts with the first element of this
array.n * n * @param [operation] function that takes the index of an element, current accumulator value and
the element itself, and calculates the next accumulator value.n * n * @sample
samples.collections.Collections.Aggregates.runningReduce

```

```

*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
CharArray.runningReduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): List<Char> {\n  if (isEmpty())
return emptyList()\n  var accumulator = this[0]\n  val result = ArrayList<Char>(size).apply { add(accumulator)
}\n  for (index in 1 until size) {\n    accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element and current accumulator value that
starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current
accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <T, R>
Array<out T>.scan(initial: R, operation: (acc: R, T) -> R): List<R> {\n  return runningFold(initial,
operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation]
from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that
`acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in
resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and
calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <R> ByteArray.scan(initial: R, operation: (acc: R, Byte) -> R): List<R> {\n  return runningFold(initial,
operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation]
from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that
`acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in
resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and
calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <R> ShortArray.scan(initial: R, operation: (acc: R, Short) -> R): List<R> {\n  return
runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by
applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial]
value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would
affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value
and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <R> IntArray.scan(initial: R, operation: (acc: R, Int) -> R): List<R> {\n  return runningFold(initial,
operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation]
from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that
`acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in
resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and
calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <R> LongArray.scan(initial: R, operation: (acc: R, Long) -> R): List<R> {\n  return
runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by
applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial]
value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would
affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value
and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic

```



```

c inline fun <R> FloatArray.scan(initial: R, operation: (acc: R, Float) -> R): List<R> {\n  return
runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by
applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial]
value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would
affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value
and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
c inline fun <R> DoubleArray.scan(initial: R, operation: (acc: R, Double) -> R): List<R> {\n  return
runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by
applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial]
value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would
affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value
and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
c inline fun <R> BooleanArray.scan(initial: R, operation: (acc: R, Boolean) -> R): List<R> {\n  return
runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by
applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial]
value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would
affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value
and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
c inline fun <R> CharArray.scan(initial: R, operation: (acc: R, Char) -> R): List<R> {\n  return runningFold(initial,
operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation]
from left to right\n * to each element, its index in the original array and current accumulator value that starts with
[initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it
would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an
element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n *
@sample samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <T, R>
Array<out T>.scanIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {\n  return
runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values
generated by applying [operation] from left to right\n * to each element, its index in the original array and current
accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should
not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
c inline fun <R> ByteArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Byte) -> R): List<R> {\n  return
runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values
generated by applying [operation] from left to right\n * to each element, its index in the original array and current
accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should
not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic

```

```

c inline fun <R> ShortArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): List<R> {\n  return
runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values
generated by applying [operation] from left to right\n * to each element, its index in the original array and current
accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should
not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
c inline fun <R> IntArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): List<R> {\n  return
runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values
generated by applying [operation] from left to right\n * to each element, its index in the original array and current
accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should
not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
c inline fun <R> LongArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Long) -> R): List<R> {\n  return
runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values
generated by applying [operation] from left to right\n * to each element, its index in the original array and current
accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should
not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
c inline fun <R> FloatArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Float) -> R): List<R> {\n  return
runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values
generated by applying [operation] from left to right\n * to each element, its index in the original array and current
accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should
not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
c inline fun <R> DoubleArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Double) -> R): List<R> {\n  return
runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values
generated by applying [operation] from left to right\n * to each element, its index in the original array and current
accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should
not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
c inline fun <R> BooleanArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Boolean) -> R): List<R> {\n  return
runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values
generated by applying [operation] from left to right\n * to each element, its index in the original array and current
accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should
not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic

```

```

c inline fun <R> CharArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): List<R> {\n  return
runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n */\n@Deprecated("Use sumOf instead.\",
ReplaceWith("this.sumOf(selector)\")\n)\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun <T>
Array<out T>.sumBy(selector: (T) -> Int): Int {\n  var sum: Int = 0\n  for (element in this) {\n    sum +=
selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n */\n@Deprecated("Use sumOf instead.\",
ReplaceWith("this.sumOf(selector)\")\n)\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
ByteArray.sumBy(selector: (Byte) -> Int): Int {\n  var sum: Int = 0\n  for (element in this) {\n    sum +=
selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n */\n@Deprecated("Use sumOf instead.\",
ReplaceWith("this.sumOf(selector)\")\n)\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
ShortArray.sumBy(selector: (Short) -> Int): Int {\n  var sum: Int = 0\n  for (element in this) {\n    sum +=
selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n */\n@Deprecated("Use sumOf instead.\",
ReplaceWith("this.sumOf(selector)\")\n)\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
IntArray.sumBy(selector: (Int) -> Int): Int {\n  var sum: Int = 0\n  for (element in this) {\n    sum +=
selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n */\n@Deprecated("Use sumOf instead.\",
ReplaceWith("this.sumOf(selector)\")\n)\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
LongArray.sumBy(selector: (Long) -> Int): Int {\n  var sum: Int = 0\n  for (element in this) {\n    sum +=
selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n */\n@Deprecated("Use sumOf instead.\",
ReplaceWith("this.sumOf(selector)\")\n)\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
FloatArray.sumBy(selector: (Float) -> Int): Int {\n  var sum: Int = 0\n  for (element in this) {\n    sum +=
selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n */\n@Deprecated("Use sumOf instead.\",
ReplaceWith("this.sumOf(selector)\")\n)\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
DoubleArray.sumBy(selector: (Double) -> Int): Int {\n  var sum: Int = 0\n  for (element in this) {\n    sum +=
selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n */\n@Deprecated("Use sumOf instead.\",
ReplaceWith("this.sumOf(selector)\")\n)\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
BooleanArray.sumBy(selector: (Boolean) -> Int): Int {\n  var sum: Int = 0\n  for (element in this) {\n    sum +=
selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n */\n@Deprecated("Use sumOf instead.\",
ReplaceWith("this.sumOf(selector)\")\n)\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
CharArray.sumBy(selector: (Char) -> Int): Int {\n  var sum: Int = 0\n  for (element in this) {\n    sum +=
selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n */\n@Deprecated("Use sumOf instead.\",
ReplaceWith("this.sumOf(selector)\")\n)\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun <T>
Array<out T>.sumByDouble(selector: (T) -> Double): Double {\n  var sum: Double = 0.0\n  for (element in this)
{\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n */\n@Deprecated("Use sumOf instead.\",
ReplaceWith("this.sumOf(selector)\")\n)\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
ByteArray.sumByDouble(selector: (Byte) -> Double): Double {\n  var sum: Double = 0.0\n  for (element in this)
{\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n */\n@Deprecated("Use sumOf instead.\",
ReplaceWith("this.sumOf(selector)\")\n)\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun

```

```

ShortArray.sumByDouble(selector: (Short) -> Double): Double {
    var sum: Double = 0.0
    for (element in this)
        sum += selector(element)
    return sum
}

/** Returns the sum of all values produced by
[selector] function applied to each element in the array.
*/
@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")
@DeprecatedSinceKotlin(warningSince = "1.5")
public inline fun

IntArray.sumByDouble(selector: (Int) -> Double): Double {
    var sum: Double = 0.0
    for (element in this)
        sum += selector(element)
    return sum
}

/** Returns the sum of all values produced by [selector]
function applied to each element in the array.
*/
@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")
@DeprecatedSinceKotlin(warningSince = "1.5")
public inline fun

LongArray.sumByDouble(selector: (Long) -> Double): Double {
    var sum: Double = 0.0
    for (element in this)
        sum += selector(element)
    return sum
}

/** Returns the sum of all values produced by
[selector] function applied to each element in the array.
*/
@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")
@DeprecatedSinceKotlin(warningSince = "1.5")
public inline fun

FloatArray.sumByDouble(selector: (Float) -> Double): Double {
    var sum: Double = 0.0
    for (element in this)
        sum += selector(element)
    return sum
}

/** Returns the sum of all values produced by
[selector] function applied to each element in the array.
*/
@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")
@DeprecatedSinceKotlin(warningSince = "1.5")
public inline fun

DoubleArray.sumByDouble(selector: (Double) -> Double): Double {
    var sum: Double = 0.0
    for (element in
this)
        sum += selector(element)
    return sum
}

/** Returns the sum of all values produced by
[selector] function applied to each element in the array.
*/
@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")
@DeprecatedSinceKotlin(warningSince = "1.5")
public inline fun

BooleanArray.sumByDouble(selector: (Boolean) -> Double): Double {
    var sum: Double = 0.0
    for (element
in this)
        sum += selector(element)
    return sum
}

/** Returns the sum of all values produced
by [selector] function applied to each element in the array.
*/
@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")
@DeprecatedSinceKotlin(warningSince = "1.5")
public inline fun

CharArray.sumByDouble(selector: (Char) -> Double): Double {
    var sum: Double = 0.0
    for (element in this)
        sum += selector(element)
    return sum
}

/** Returns the sum of all values produced by
[selector] function applied to each element in the array.
*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("sumOfDouble")
@kotlin.internal.InlineOnly
public inline fun
<T> Array<out T>.sumOf(selector: (T) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element
in this)
        sum += selector(element)
    return sum
}

/** Returns the sum of all values produced
by [selector] function applied to each element in the array.
*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("sumOfDouble")
@kotlin.internal.InlineOnly
public inline fun

ByteArray.sumOf(selector: (Byte) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in
this)
        sum += selector(element)
    return sum
}

/** Returns the sum of all values produced by
[selector] function applied to each element in the array.
*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("sumOfDouble")
@kotlin.internal.InlineOnly
public inline fun

ShortArray.sumOf(selector: (Short) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in
this)
        sum += selector(element)
    return sum
}

/** Returns the sum of all values produced by
[selector] function applied to each element in the array.
*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("sumOfDouble")
@kotlin.internal.InlineOnly
public inline fun

IntArray.sumOf(selector: (Int) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this)
        sum += selector(element)
    return sum
}

/** Returns the sum of all values produced by
[selector] function applied to each element in the array.
*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution

```

```

ByLambdaReturnType\n@kotlin.jvm.JvmName(\sumOfDouble)\n@kotlin.internal.InlineOnly\npublic inline fun
LongArray.sumOf(selector: (Long) -> Double): Double {\n  var sum: Double = 0.toDouble()\n  for (element in
this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@SinceKotlin(\1.4)\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\sumOfDouble)\n@kotlin.internal.InlineOnly\npublic inline fun
FloatArray.sumOf(selector: (Float) -> Double): Double {\n  var sum: Double = 0.toDouble()\n  for (element in
this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@SinceKotlin(\1.4)\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\sumOfDouble)\n@kotlin.internal.InlineOnly\npublic inline fun
DoubleArray.sumOf(selector: (Double) -> Double): Double {\n  var sum: Double = 0.toDouble()\n  for (element
in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced
by [selector] function applied to each element in the array.\n
*\n@SinceKotlin(\1.4)\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\sumOfDouble)\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.sumOf(selector: (Boolean) -> Double): Double {\n  var sum: Double = 0.toDouble()\n  for
(element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values
produced by [selector] function applied to each element in the array.\n
*\n@SinceKotlin(\1.4)\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\sumOfDouble)\n@kotlin.internal.InlineOnly\npublic inline fun
CharArray.sumOf(selector: (Char) -> Double): Double {\n  var sum: Double = 0.toDouble()\n  for (element in
this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@SinceKotlin(\1.4)\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\sumOfInt)\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Array<out T>.sumOf(selector: (T) -> Int): Int {\n  var sum: Int = 0.toInt()\n  for (element in this) {\n    sum +=
selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n
*\n@SinceKotlin(\1.4)\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\sumOfInt)\n@kotlin.internal.InlineOnly\npublic inline fun
ByteArray.sumOf(selector: (Byte) -> Int): Int {\n  var sum: Int = 0.toInt()\n  for (element in this) {\n    sum +=
selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n
*\n@SinceKotlin(\1.4)\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\sumOfInt)\n@kotlin.internal.InlineOnly\npublic inline fun
ShortArray.sumOf(selector: (Short) -> Int): Int {\n  var sum: Int = 0.toInt()\n  for (element in this) {\n    sum +=
selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n
*\n@SinceKotlin(\1.4)\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\sumOfInt)\n@kotlin.internal.InlineOnly\npublic inline fun
IntArray.sumOf(selector: (Int) -> Int): Int {\n  var sum: Int = 0.toInt()\n  for (element in this) {\n    sum +=
selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n
*\n@SinceKotlin(\1.4)\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\sumOfInt)\n@kotlin.internal.InlineOnly\npublic inline fun
LongArray.sumOf(selector: (Long) -> Int): Int {\n  var sum: Int = 0.toInt()\n  for (element in this) {\n    sum +=
selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function

```

applied to each element in the array.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun\nFloatArray.sumOf(selector: (Float) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum +=\n        selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function\n    applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun\nDoubleArray.sumOf(selector: (Double) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]\n    function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun\nBooleanArray.sumOf(selector: (Boolean) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]\n    function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun\nCharArray.sumOf(selector: (Char) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum +=\n        selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function\n    applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun\n<T> Array<out T>.sumOf(selector: (T) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this)\n    {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by\n    [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun\nByteArray.sumOf(selector: (Byte) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]\n    function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun\nShortArray.sumOf(selector: (Short) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]\n    function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun\nIntArray.sumOf(selector: (Int) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]\n    function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun\nLongArray.sumOf(selector: (Long) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]\n    function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun
```

```

FloatArray.sumOf(selector: (Float) -> Long): Long {\n  var sum: Long = 0.toLong()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]\n function applied to each element in the array.\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun\nDoubleArray.sumOf(selector: (Double) -> Long): Long {\n  var sum: Long = 0.toLong()\n  for (element in this)\n  {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by\n [selector] function applied to each element in the array.\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun\nBooleanArray.sumOf(selector: (Boolean) -> Long): Long {\n  var sum: Long = 0.toLong()\n  for (element in this)\n  {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by\n [selector] function applied to each element in the array.\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun\nCharArray.sumOf(selector: (Char) -> Long): Long {\n  var sum: Long = 0.toLong()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]\n function applied to each element in the array.\n\n*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType\ns::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.sumOf(selector: (T) -> UInt): UInt {\n  var sum: UInt = 0.toUInt()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return\n  sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the\n array.\n\n*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType\ns::class)\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.sumOf(selector: (Byte) -> UInt): UInt {\n  var\n  sum: UInt = 0.toUInt()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n\n*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType\ns::class)\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.sumOf(selector: (Short) -> UInt): UInt {\n  var\n  sum: UInt = 0.toUInt()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n\n*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType\ns::class)\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.sumOf(selector: (Int) -> UInt): UInt {\n  var\n  sum: UInt = 0.toUInt()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n\n*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType\ns::class)\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.sumOf(selector: (Long) -> UInt): UInt {\n  var\n  sum: UInt = 0.toUInt()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n\n*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType\ns::class)\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.sumOf(selector: (Float) -> UInt): UInt {\n  var\n  sum: UInt = 0.toUInt()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n\n*/

```

Returns the sum of all values produced by [selector] function applied to each element in the array.\n

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType\ns::class)\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.sumOf(selector: (Double) -> UInt): UInt {\n\n    var sum: UInt = 0.toUInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return\n    sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the\n    array.\n */
```

Returns the sum of all values produced by [selector] function applied to each element in the array.\n

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType\ns::class)\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.sumOf(selector: (Boolean) -> UInt): UInt\n{\n\n    var sum: UInt = 0.toUInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return\n    sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the\n    array.\n */
```

Returns the sum of all values produced by [selector] function applied to each element in the array.\n

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType\ns::class)\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.sumOf(selector: (Char) -> UInt): UInt {\n\n    var\n    sum: UInt = 0.toUInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the\n    array.\n */
```

Returns the sum of all values produced by [selector] function applied to each element in the array.\n

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy\npes::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.sumOf(selector: (T) -> ULong):\n    ULong {\n\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in\n    the array.\n */
```

Returns the sum of all values produced by [selector] function applied to each element in the array.\n

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy\npes::class)\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.sumOf(selector: (Byte) -> ULong): ULong\n{\n\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return\n    sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the\n    array.\n */
```

Returns the sum of all values produced by [selector] function applied to each element in the array.\n

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy\npes::class)\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.sumOf(selector: (Short) -> ULong): ULong\n{\n\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return\n    sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the\n    array.\n */
```

Returns the sum of all values produced by [selector] function applied to each element in the array.\n

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy\npes::class)\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.sumOf(selector: (Int) -> ULong): ULong {\n\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return\n    sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the\n    array.\n */
```

Returns the sum of all values produced by [selector] function applied to each element in the array.\n

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy\npes::class)\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.sumOf(selector: (Long) -> ULong): ULong\n{\n\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return\n    sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the\n    array.\n */
```



```

*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.sumOf(selector: (Float) -> ULong): ULong
{\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the
array.\n

```

```

*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.sumOf(selector: (Double) -> ULong):
ULong {\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n
return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in
the array.\n

```

```

*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.sumOf(selector: (Boolean) -> ULong):
ULong {\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n
return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in
the array.\n

```

```

*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.sumOf(selector: (Char) -> ULong): ULong
{\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return
sum\n}\n\n/**\n * Returns an original collection containing all the non-`null` elements, throwing an
[IllegalArgumentException] if there are any `null` elements.\n\npublic fun <T : Any>
Array<T?>.requireNonNulls(): Array<T> {\n    for (element in this) {\n        if (element == null) {\n            throw
IllegalArgumentException("null element found in $this.")\n        }\n    }\n}

```

```

@Suppress("UNCHECKED_CAST")\n    return this as Array<T>\n}\n\n/**\n * Splits the original array into pair
of lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n * while *second* list contains
elements for which [predicate] yielded `false`.\n * \n * @sample

```

```

samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n\npublic inline fun <T> Array<out
T>.partition(predicate: (T) -> Boolean): Pair<List<T>, List<T>> {\n    val first = ArrayList<T>()\n    val second =
ArrayList<T>()\n    for (element in this) {\n        if (predicate(element)) {\n            first.add(element)\n        }
else {\n            second.add(element)\n        }\n    }\n    return Pair(first, second)\n}\n\n/**\n * Splits the original array into pair
of lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n * while *second* list contains
elements for which [predicate] yielded `false`.\n * \n * @sample

```

```

samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n\npublic inline fun
ByteArray.partition(predicate: (Byte) -> Boolean): Pair<List<Byte>, List<Byte>> {\n    val first =
ArrayList<Byte>()\n    val second = ArrayList<Byte>()\n    for (element in this) {\n        if (predicate(element)) {\n
            first.add(element)\n        } else {\n            second.add(element)\n        }\n    }\n    return Pair(first,
second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which
[predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n *
@sample samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n\npublic inline fun
ShortArray.partition(predicate: (Short) -> Boolean): Pair<List<Short>, List<Short>> {\n    val first =
ArrayList<Short>()\n    val second = ArrayList<Short>()\n    for (element in this) {\n        if (predicate(element)) {\n
            first.add(element)\n        } else {\n            second.add(element)\n        }\n    }\n    return Pair(first,
second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which
[predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n *
@sample samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n\npublic inline fun

```

```

IntArray.partition(predicate: (Int) -> Boolean): Pair<List<Int>, List<Int>> {\n  val first = ArrayList<Int>()\n  val second = ArrayList<Int>()\n  for (element in this) {\n    if (predicate(element)) {\n      first.add(element)\n    } else {\n      second.add(element)\n    }\n  }\n  return Pair(first, second)\n}\n\n/*\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * @sample
samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun
LongArray.partition(predicate: (Long) -> Boolean): Pair<List<Long>, List<Long>> {\n  val first =
ArrayList<Long>()\n  val second = ArrayList<Long>()\n  for (element in this) {\n    if (predicate(element)) {\n
      first.add(element)\n    } else {\n      second.add(element)\n    }\n  }\n  return Pair(first,
second)\n}\n\n/*\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which
[predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * @sample
samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun
FloatArray.partition(predicate: (Float) -> Boolean): Pair<List<Float>, List<Float>> {\n  val first =
ArrayList<Float>()\n  val second = ArrayList<Float>()\n  for (element in this) {\n    if (predicate(element)) {\n
      first.add(element)\n    } else {\n      second.add(element)\n    }\n  }\n  return Pair(first,
second)\n}\n\n/*\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which
[predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * @sample
samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun
DoubleArray.partition(predicate: (Double) -> Boolean): Pair<List<Double>, List<Double>> {\n  val first =
ArrayList<Double>()\n  val second = ArrayList<Double>()\n  for (element in this) {\n    if (predicate(element))
{\n      first.add(element)\n    } else {\n      second.add(element)\n    }\n  }\n  return Pair(first,
second)\n}\n\n/*\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which
[predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * @sample
samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun
BooleanArray.partition(predicate: (Boolean) -> Boolean): Pair<List<Boolean>, List<Boolean>> {\n  val first =
ArrayList<Boolean>()\n  val second = ArrayList<Boolean>()\n  for (element in this) {\n    if
(predicate(element)) {\n      first.add(element)\n    } else {\n      second.add(element)\n    }\n  }\n  return Pair(first, second)\n}\n\n/*\n * Splits the original array into pair of lists,\n * where *first* list contains
elements for which [predicate] yielded `true`,\n * while *second* list contains elements for which [predicate]
yielded `false`.\n * @sample
samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun
CharArray.partition(predicate: (Char) -> Boolean): Pair<List<Char>, List<Char>> {\n  val first =
ArrayList<Char>()\n  val second = ArrayList<Char>()\n  for (element in this) {\n    if
(predicate(element)) {\n      first.add(element)\n    } else {\n      second.add(element)\n    }\n  }\n  return Pair(first, second)\n}\n\n/*\n * Returns a list of pairs built from the elements of `this` array and the [other]
array with the same index.\n * The returned list has length of the shortest collection.\n * @sample
samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <T, R> Array<out T>.zip(other:
Array<out R>): List<Pair<T, R>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/*\n * Returns a list of pairs
built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of
the shortest collection.\n * @sample
samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> ByteArray.zip(other: Array<out R>): List<Pair<Byte, R>> {\n  return zip(other) { t1, t2 -> t1 to t2
}\n}\n\n/*\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same
index.\n * The returned list has length of the shortest collection.\n * @sample
samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> ShortArray.zip(other: Array<out
R>): List<Pair<Short, R>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/*\n * Returns a list of pairs built from
the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest
collection.\n * @sample
samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R>
IntArray.zip(other: Array<out R>): List<Pair<Int, R>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/*\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The

```

```

returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n * \n public infix fun <R> LongArray.zip(other: Array<out
R>): List<Pair<Long, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from
the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n * \n public infix fun <R>
FloatArray.zip(other: Array<out R>): List<Pair<Float, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n *
Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The
returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n * \n public infix fun <R> DoubleArray.zip(other: Array<out
R>): List<Pair<Double, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built
from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the
shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n * \n public infix fun <R>
BooleanArray.zip(other: Array<out R>): List<Pair<Boolean, R>> {\n    return zip(other) { t1, t2 -> t1 to t2
}\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same
index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n * \n public infix fun <R> CharArray.zip(other: Array<out R>):
List<Pair<Char, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of values built from the
elements of `this` array and the [other] array with the same index\n * using the provided [transform] function
applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n * \n public inline fun <T, R, V> Array<out
T>.zip(other: Array<out R>, transform: (a: T, b: R) -> V): List<V> {\n    val size = minOf(size, other.size)\n    val
list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return
list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same
index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length
of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n * \n
public inline fun <R, V> ByteArray.zip(other: Array<out R>, transform: (a: Byte, b: R) -> V): List<V> {\n    val
size = minOf(size, other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements
of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each
pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n * \n public inline fun <R, V>
ShortArray.zip(other: Array<out R>, transform: (a: Short, b: R) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n * \n public inline fun <R, V>
IntArray.zip(other: Array<out R>, transform: (a: Int, b: R) -> V): List<V> {\n    val size = minOf(size, other.size)\n
val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return
list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same
index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length
of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n * \n
public inline fun <R, V> LongArray.zip(other: Array<out R>, transform: (a: Long, b: R) -> V): List<V> {\n    val
size = minOf(size, other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements
of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each
pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n * \n public inline fun <R, V>

```

```

FloatArray.zip(other: Array<out R>, transform: (a: Float, b: R) -> V): List<V> {\n  val size = minOf(size,
other.size)\n  val list = ArrayList<V>(size)\n  for (i in 0 until size) {\n    list.add(transform(this[i], other[i]))\n
}\n  return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n *\npublic inline fun <R, V>
DoubleArray.zip(other: Array<out R>, transform: (a: Double, b: R) -> V): List<V> {\n  val size = minOf(size,
other.size)\n  val list = ArrayList<V>(size)\n  for (i in 0 until size) {\n    list.add(transform(this[i], other[i]))\n
}\n  return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n *\npublic inline fun <R, V>
BooleanArray.zip(other: Array<out R>, transform: (a: Boolean, b: R) -> V): List<V> {\n  val size = minOf(size,
other.size)\n  val list = ArrayList<V>(size)\n  for (i in 0 until size) {\n    list.add(transform(this[i], other[i]))\n
}\n  return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n *\npublic inline fun <R, V>
CharArray.zip(other: Array<out R>, transform: (a: Char, b: R) -> V): List<V> {\n  val size = minOf(size,
other.size)\n  val list = ArrayList<V>(size)\n  for (i in 0 until size) {\n    list.add(transform(this[i], other[i]))\n
}\n  return list\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with
the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n *\npublic infix fun <T, R> Array<out T>.zip(other:
Iterable<R>): List<Pair<T, R>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built
from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the
shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n *\npublic infix fun <R>
ByteArray.zip(other: Iterable<R>): List<Pair<Byte, R>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n *
Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The
returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n *\npublic infix fun <R> ShortArray.zip(other: Iterable<R>):
List<Pair<Short, R>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the
elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n *\npublic infix fun <R>
IntArray.zip(other: Iterable<R>): List<Pair<Int, R>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n *
Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The
returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n *\npublic infix fun <R> LongArray.zip(other: Iterable<R>):
List<Pair<Long, R>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the
elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n *\npublic infix fun <R>
FloatArray.zip(other: Iterable<R>): List<Pair<Float, R>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n *
Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The
returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n *\npublic infix fun <R> DoubleArray.zip(other:
Iterable<R>): List<Pair<Double, R>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs
built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of
the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n *\npublic infix fun
<R> BooleanArray.zip(other: Iterable<R>): List<Pair<Boolean, R>> {\n  return zip(other) { t1, t2 -> t1 to t2

```

```

}
}

Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
The returned list has length of the shortest collection.
@sample
samples.collections.Iterables.Operations.zipIterable
public infix fun <R> CharArray.zip(other: Iterable<R>): List<Pair<Char, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

Returns a list of values built from the elements of `this` array and the [other] collection with the same index
using the provided [transform] function applied to each pair of elements.
The returned list has length of the shortest collection.
@sample
samples.collections.Iterables.Operations.zipIterableWithTransform
public inline fun <T, R, V> Array<out T>.zip(other: Iterable<R>, transform: (a: T, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

Returns a list of values built from the elements of `this` array and the [other] collection with the same index
using the provided [transform] function applied to each pair of elements.
The returned list has length of the shortest collection.
@sample
samples.collections.Iterables.Operations.zipIterableWithTransform
public inline fun <R, V> ByteArray.zip(other: Iterable<R>, transform: (a: Byte, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

Returns a list of values built from the elements of `this` array and the [other] collection with the same index
using the provided [transform] function applied to each pair of elements.
The returned list has length of the shortest collection.
@sample
samples.collections.Iterables.Operations.zipIterableWithTransform
public inline fun <R, V> ShortArray.zip(other: Iterable<R>, transform: (a: Short, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

Returns a list of values built from the elements of `this` array and the [other] collection with the same index
using the provided [transform] function applied to each pair of elements.
The returned list has length of the shortest collection.
@sample
samples.collections.Iterables.Operations.zipIterableWithTransform
public inline fun <R, V> IntArray.zip(other: Iterable<R>, transform: (a: Int, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

Returns a list of values built from the elements of `this` array and the [other] collection with the same index
using the provided [transform] function applied to each pair of elements.
The returned list has length of the shortest collection.
@sample
samples.collections.Iterables.Operations.zipIterableWithTransform
public inline fun <R, V> LongArray.zip(other: Iterable<R>, transform: (a: Long, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

Returns a list of values built from the elements of `this` array and the [other] collection with the same index
using the provided [transform] function applied to each pair of elements.
The returned list has length of the shortest collection.
@sample
samples.collections.Iterables.Operations.zipIterableWithTransform
public inline fun <R, V> FloatArray.zip(other: Iterable<R>, transform: (a: Float, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

Returns a list of values built from the elements of `this` array and the [other] collection with the same index
using the provided [transform] function applied to each pair of elements.
The returned list has length of the shortest collection.
@sample
samples.collections.Iterables.Operations.zipIterableWithTransform
public inline fun <R, V> DoubleArray.zip(other: Iterable<R>, transform: (a: Double, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

```

```

list\}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with the
same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has
length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n *^\npublic inline fun <R, V>
BooleanArray.zip(other: Iterable<R>, transform: (a: Boolean, b: R) -> V): List<V> {\n    val arraySize = size\n    val
list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n    var i = 0\n    for (element in other)
{\n        if (i >= arraySize) break\n        list.add(transform(this[i++], element))\n    }\n    return list\}\n\n/**\n *
Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n *
using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the
shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*^\npublic inline fun <R, V> CharArray.zip(other: Iterable<R>, transform: (a: Char, b: R) -> V): List<V> {\n    val
arraySize = size\n    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n    var i = 0\n
for (element in other) {\n        if (i >= arraySize) break\n        list.add(transform(this[i++], element))\n    }\n
return list\}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same
index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n *^\npublic infix fun ByteArray.zip(other: ByteArray):
List<Pair<Byte, Byte>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from
the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n *^\npublic infix fun
ShortArray.zip(other: ShortArray): List<Pair<Short, Short>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n
* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The
returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n *^\npublic infix fun IntArray.zip(other: IntArray):
List<Pair<Int, Int>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the
elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n *^\npublic infix fun
LongArray.zip(other: LongArray): List<Pair<Long, Long>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n
* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The
returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n *^\npublic infix fun FloatArray.zip(other: FloatArray):
List<Pair<Float, Float>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from
the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n *^\npublic infix fun
DoubleArray.zip(other: DoubleArray): List<Pair<Double, Double>> {\n    return zip(other) { t1, t2 -> t1 to t2
}\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same
index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n *^\npublic infix fun BooleanArray.zip(other: BooleanArray):
List<Pair<Boolean, Boolean>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built
from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the
shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n *^\npublic infix fun
CharArray.zip(other: CharArray): List<Pair<Char, Char>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n
* Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the
provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest
array.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n *^\npublic inline fun
<V> ByteArray.zip(other: ByteArray, transform: (a: Byte, b: Byte) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n
}\n    return list\}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned

```

```

list has length of the shortest array.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n *\npublic inline fun <V>
ShortArray.zip(other: ShortArray, transform: (a: Short, b: Short) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n
}\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest array.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n *\npublic inline fun <V> IntArray.zip(other:
IntArray, transform: (a: Int, b: Int) -> V): List<V> {\n    val size = minOf(size, other.size)\n    val list =
ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return
list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same
index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length
of the shortest array.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\npublic inline fun <V> LongArray.zip(other: LongArray, transform: (a: Long, b: Long) -> V): List<V> {\n    val
size = minOf(size, other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n
list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements
of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each
pair of elements.\n * The returned list has length of the shortest array.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n *\npublic inline fun <V>
FloatArray.zip(other: FloatArray, transform: (a: Float, b: Float) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n
}\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest array.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n *\npublic inline fun <V>
DoubleArray.zip(other: DoubleArray, transform: (a: Double, b: Double) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n
}\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest array.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n *\npublic inline fun <V>
BooleanArray.zip(other: BooleanArray, transform: (a: Boolean, b: Boolean) -> V): List<V> {\n    val size =
minOf(size, other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i],
other[i]))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the
[other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n *
The returned list has length of the shortest array.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n *\npublic inline fun <V>
CharArray.zip(other: CharArray, transform: (a: Char, b: Char) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n
}\n    return list\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the
given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value
of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string
(which defaults to "...").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n *\npublic fun
<T, A : Appendable> Array<out T>.joinTo(buffer: A, separator: CharSequence = '\n', prefix: CharSequence = "\n",
postfix: CharSequence = "\n", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? =
null): A {\n    buffer.append(prefix)\n    var count = 0\n    for (element in this) {\n        if (++count > 1)
buffer.append(separator)\n        if (limit < 0 || count <= limit) {\n            buffer.appendElement(element, transform)\n
        } else break\n    }\n    if (limit >= 0 && count > limit) buffer.append(truncated)\n    buffer.append(postfix)\n
}

```



```

buffer.append(transform(element))\n      else\n          buffer.append(element.toString())\n      } else break\n
}\n if (limit >= 0 && count > limit) buffer.append(truncated)\n buffer.append(postfix)\n return
buffer\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given
[prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of
[limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which
defaults to "...").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n */\npublic fun <A :
Appendable> DoubleArray.joinTo(buffer: A, separator: CharSequence = "\", \"", prefix: CharSequence = "\"\", postfix:
CharSequence = "\"", limit: Int = -1, truncated: CharSequence = "...\", transform: ((Double) -> CharSequence)? =
null): A {\n    buffer.append(prefix)\n    var count = 0\n    for (element in this) {\n        if (++count > 1)
buffer.append(separator)\n        if (limit < 0 || count <= limit) {\n            if (transform != null)\n                buffer.append(transform(element))\n            else\n                buffer.append(element.toString())\n        } else break\n
}\n    if (limit >= 0 && count > limit) buffer.append(truncated)\n    buffer.append(postfix)\n    return
buffer\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given
[prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of
[limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which
defaults to "...").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n */\npublic fun <A :
Appendable> BooleanArray.joinTo(buffer: A, separator: CharSequence = "\", \"", prefix: CharSequence = "\"\", postfix:
CharSequence = "\"", limit: Int = -1, truncated: CharSequence = "...\", transform: ((Boolean) ->
CharSequence)? = null): A {\n    buffer.append(prefix)\n    var count = 0\n    for (element in this) {\n        if (++count
> 1) buffer.append(separator)\n        if (limit < 0 || count <= limit) {\n            if (transform != null)\n                buffer.append(transform(element))\n            else\n                buffer.append(element.toString())\n        } else break\n
}\n    if (limit >= 0 && count > limit) buffer.append(truncated)\n    buffer.append(postfix)\n    return
buffer\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given
[prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of
[limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which
defaults to "...").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n */\npublic fun <A :
Appendable> CharArray.joinTo(buffer: A, separator: CharSequence = "\", \"", prefix: CharSequence = "\"\", postfix:
CharSequence = "\"", limit: Int = -1, truncated: CharSequence = "...\", transform: ((Char) -> CharSequence)? =
null): A {\n    buffer.append(prefix)\n    var count = 0\n    for (element in this) {\n        if (++count > 1)
buffer.append(separator)\n        if (limit < 0 || count <= limit) {\n            if (transform != null)\n                buffer.append(transform(element))\n            else\n                buffer.append(element)\n        } else break\n
}\n    if (limit >= 0 && count > limit) buffer.append(truncated)\n    buffer.append(postfix)\n    return buffer\n}\n\n\n/**\n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if
supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only
the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to "...").\n * \n *
@sample samples.collections.Collections.Transformations.joinToString\n */\npublic fun <T> Array<out
T>.joinToString(separator: CharSequence = "\", \"", prefix: CharSequence = "\"\", postfix: CharSequence = "\"", limit:
Int = -1, truncated: CharSequence = "...\", transform: ((T) -> CharSequence)? = null): String {\n    return
joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n\n/**\n * Creates a
string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n
* If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first
[limit]\n * elements will be appended, followed by the [truncated] string (which defaults to "...").\n * \n *
@sample samples.collections.Collections.Transformations.joinToString\n */\npublic fun ByteArray.joinToString(separator:
CharSequence = "\", \"", prefix: CharSequence = "\"\", postfix: CharSequence = "\"", limit: Int = -1, truncated:
CharSequence = "...\", transform: ((Byte) -> CharSequence)? = null): String {\n    return joinTo(StringBuilder(),
separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n\n/**\n * Creates a string from all the elements
separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be
huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be

```

appended, followed by the [truncated] string (which defaults to "...").

```

samples.collections.Collections.Transformations.joinToString\n *^npublic fun ShortArray.joinToString(separator: CharSequence = "\", \"", prefix: CharSequence = "\", postfix: CharSequence = "\", limit: Int = -1, truncated: CharSequence = "...\", transform: ((Short) -> CharSequence)? = null): String {\n    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n**\n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to "...").\n * \n * @sample

```

appended, followed by the [truncated] string (which defaults to "...").

```

samples.collections.Collections.Transformations.joinToString\n *^npublic fun IntArray.joinToString(separator: CharSequence = "\", \"", prefix: CharSequence = "\", postfix: CharSequence = "\", limit: Int = -1, truncated: CharSequence = "...\", transform: ((Int) -> CharSequence)? = null): String {\n    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n**\n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to "...").\n * \n * @sample

```

appended, followed by the [truncated] string (which defaults to "...").

```

samples.collections.Collections.Transformations.joinToString\n *^npublic fun LongArray.joinToString(separator: CharSequence = "\", \"", prefix: CharSequence = "\", postfix: CharSequence = "\", limit: Int = -1, truncated: CharSequence = "...\", transform: ((Long) -> CharSequence)? = null): String {\n    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n**\n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to "...").\n * \n * @sample

```

appended, followed by the [truncated] string (which defaults to "...").

```

samples.collections.Collections.Transformations.joinToString\n *^npublic fun FloatArray.joinToString(separator: CharSequence = "\", \"", prefix: CharSequence = "\", postfix: CharSequence = "\", limit: Int = -1, truncated: CharSequence = "...\", transform: ((Float) -> CharSequence)? = null): String {\n    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n**\n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to "...").\n * \n * @sample

```

appended, followed by the [truncated] string (which defaults to "...").

```

samples.collections.Collections.Transformations.joinToString\n *^npublic fun DoubleArray.joinToString(separator: CharSequence = "\", \"", prefix: CharSequence = "\", postfix: CharSequence = "\", limit: Int = -1, truncated: CharSequence = "...\", transform: ((Double) -> CharSequence)? = null): String {\n    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n**\n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to "...").\n * \n * @sample

```

appended, followed by the [truncated] string (which defaults to "...").

```

samples.collections.Collections.Transformations.joinToString\n *^npublic fun BooleanArray.joinToString(separator: CharSequence = "\", \"", prefix: CharSequence = "\", postfix: CharSequence = "\", limit: Int = -1, truncated: CharSequence = "...\", transform: ((Boolean) -> CharSequence)? = null): String {\n    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n**\n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to "...").\n * \n * @sample

```

appended, followed by the [truncated] string (which defaults to "...").

```

samples.collections.Collections.Transformations.joinToString\n *^npublic fun CharArray.joinToString(separator: CharSequence = "\", \"", prefix: CharSequence = "\", postfix: CharSequence = "\", limit: Int = -1, truncated: CharSequence = "...\", transform: ((Char) -> CharSequence)? = null): String {\n    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n**\n * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.\n *^npublic fun <T> Array<out T>.asIterable():

```

```

Iterable<T> {\n  if (isEmpty()) return emptyList()\n  return Iterable { this.iterator() }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.\n */\npublic fun\nByteArray.asIterable(): Iterable<Byte> {\n  if (isEmpty()) return emptyList()\n  return Iterable { this.iterator()\n }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original array returning its elements when being\n iterated.\n */\npublic fun ShortArray.asIterable(): Iterable<Short> {\n  if (isEmpty()) return emptyList()\n  return\n Iterable { this.iterator() }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original array returning its\n elements when being iterated.\n */\npublic fun IntArray.asIterable(): Iterable<Int> {\n  if (isEmpty()) return\n emptyList()\n  return Iterable { this.iterator() }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original\n array returning its elements when being iterated.\n */\npublic fun LongArray.asIterable(): Iterable<Long> {\n  if\n (isEmpty()) return emptyList()\n  return Iterable { this.iterator() }\n}\n\n/**\n * Creates an [Iterable] instance that\n wraps the original array returning its elements when being iterated.\n */\npublic fun FloatArray.asIterable():\n Iterable<Float> {\n  if (isEmpty()) return emptyList()\n  return Iterable { this.iterator() }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.\n */\npublic fun\n DoubleArray.asIterable(): Iterable<Double> {\n  if (isEmpty()) return emptyList()\n  return Iterable { this.iterator()\n }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original array returning its elements when\n being iterated.\n */\npublic fun BooleanArray.asIterable(): Iterable<Boolean> {\n  if (isEmpty()) return\n emptyList()\n  return Iterable { this.iterator() }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original\n array returning its elements when being iterated.\n */\npublic fun CharArray.asIterable(): Iterable<Char> {\n  if\n (isEmpty()) return emptyList()\n  return Iterable { this.iterator() }\n}\n\n/**\n * Creates a [Sequence] instance that\n wraps the original array returning its elements when being iterated.\n */\n * \n * @sample\n samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun <T> Array<out T>.asSequence():\n Sequence<T> {\n  if (isEmpty()) return emptySequence()\n  return Sequence { this.iterator() }\n}\n\n/**\n * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n */\n * \n * @sample\n samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun ByteArray.asSequence():\n Sequence<Byte> {\n  if (isEmpty()) return emptySequence()\n  return Sequence { this.iterator() }\n}\n\n/**\n * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n */\n * \n * @sample\n samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun ShortArray.asSequence():\n Sequence<Short> {\n  if (isEmpty()) return emptySequence()\n  return Sequence { this.iterator() }\n}\n\n/**\n * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n */\n * \n * @sample\n samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun IntArray.asSequence():\n Sequence<Int> {\n  if (isEmpty()) return emptySequence()\n  return Sequence { this.iterator() }\n}\n\n/**\n * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n */\n * \n * @sample\n samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun LongArray.asSequence():\n Sequence<Long> {\n  if (isEmpty()) return emptySequence()\n  return Sequence { this.iterator() }\n}\n\n/**\n * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n */\n * \n * @sample\n samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun FloatArray.asSequence():\n Sequence<Float> {\n  if (isEmpty()) return emptySequence()\n  return Sequence { this.iterator() }\n}\n\n/**\n * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n */\n * \n * @sample\n samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun DoubleArray.asSequence():\n Sequence<Double> {\n  if (isEmpty()) return emptySequence()\n  return Sequence { this.iterator() }\n}\n\n/**\n * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n */\n * \n * @sample\n samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun\n BooleanArray.asSequence(): Sequence<Boolean> {\n  if (isEmpty()) return emptySequence()\n  return Sequence\n { this.iterator() }\n}\n\n/**\n * Creates a [Sequence] instance that wraps the original array returning its elements\n when being iterated.\n */\n * \n * @sample\n samples.collections.Sequences.Building.sequenceFromArray\n */\npublic\n fun CharArray.asSequence(): Sequence<Char> {\n  if (isEmpty()) return emptySequence()\n  return Sequence { this.iterator()\n }\n}\n\n/**\n * Returns an average value of elements in the array.\n */\n\n * \n * @kotlin.jvm.JvmName("averageOfByte")\n */\npublic fun Array<out Byte>.average(): Double {\n  var sum:

```

```

Double = 0.0\n  var count: Int = 0\n  for (element in this) {\n    sum += element\n    ++count\n  }\n  return
if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*/\n@kotlin.jvm.JvmName("averageOfShort")\npublic fun Array<out Short>.average(): Double {\n  var sum:
Double = 0.0\n  var count: Int = 0\n  for (element in this) {\n    sum += element\n    ++count\n  }\n  return
if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*/\n@kotlin.jvm.JvmName("averageOfInt")\npublic fun Array<out Int>.average(): Double {\n  var sum: Double
= 0.0\n  var count: Int = 0\n  for (element in this) {\n    sum += element\n    ++count\n  }\n  return if (count
== 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*/\n@kotlin.jvm.JvmName("averageOfLong")\npublic fun Array<out Long>.average(): Double {\n  var sum:
Double = 0.0\n  var count: Int = 0\n  for (element in this) {\n    sum += element\n    ++count\n  }\n  return
if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*/\n@kotlin.jvm.JvmName("averageOfFloat")\npublic fun Array<out Float>.average(): Double {\n  var sum:
Double = 0.0\n  var count: Int = 0\n  for (element in this) {\n    sum += element\n    ++count\n  }\n  return
if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*/\n@kotlin.jvm.JvmName("averageOfDouble")\npublic fun Array<out Double>.average(): Double {\n  var sum:
Double = 0.0\n  var count: Int = 0\n  for (element in this) {\n    sum += element\n    ++count\n  }\n  return
if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*/\npublic fun ByteArray.average(): Double {\n  var sum: Double = 0.0\n  var count: Int = 0\n  for (element in
this) {\n    sum += element\n    ++count\n  }\n  return if (count == 0) Double.NaN else sum /
count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*/\npublic fun ShortArray.average():
Double {\n  var sum: Double = 0.0\n  var count: Int = 0\n  for (element in this) {\n    sum += element\n
++count\n  }\n  return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of
elements in the array.\n
*/\npublic fun IntArray.average(): Double {\n  var sum: Double = 0.0\n  var count: Int =
0\n  for (element in this) {\n    sum += element\n    ++count\n  }\n  return if (count == 0) Double.NaN else
sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*/\npublic fun LongArray.average():
Double {\n  var sum: Double = 0.0\n  var count: Int = 0\n  for (element in this) {\n    sum += element\n
++count\n  }\n  return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of
elements in the array.\n
*/\npublic fun FloatArray.average(): Double {\n  var sum: Double = 0.0\n  var count: Int
= 0\n  for (element in this) {\n    sum += element\n    ++count\n  }\n  return if (count == 0) Double.NaN
else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*/\npublic fun
DoubleArray.average(): Double {\n  var sum: Double = 0.0\n  var count: Int = 0\n  for (element in this) {\n
sum += element\n    ++count\n  }\n  return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns
the sum of all elements in the array.\n
*/\n@kotlin.jvm.JvmName("sumOfByte")\npublic fun Array<out
Byte>.sum(): Int {\n  var sum: Int = 0\n  for (element in this) {\n    sum += element\n  }\n  return
sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*/\n@kotlin.jvm.JvmName("sumOfShort")\npublic fun Array<out Short>.sum(): Int {\n  var sum: Int = 0\n  for
(element in this) {\n    sum += element\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all elements in the
array.\n
*/\n@kotlin.jvm.JvmName("sumOfInt")\npublic fun Array<out Int>.sum(): Int {\n  var sum: Int = 0\n
for (element in this) {\n    sum += element\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all elements in
the array.\n
*/\n@kotlin.jvm.JvmName("sumOfLong")\npublic fun Array<out Long>.sum(): Long {\n  var sum:
Long = 0L\n  for (element in this) {\n    sum += element\n  }\n  return sum\n}\n\n/**\n * Returns the sum of
all elements in the array.\n
*/\n@kotlin.jvm.JvmName("sumOfFloat")\npublic fun Array<out Float>.sum(): Float
{\n  var sum: Float = 0.0f\n  for (element in this) {\n    sum += element\n  }\n  return sum\n}\n\n/**\n *
Returns the sum of all elements in the array.\n
*/\n@kotlin.jvm.JvmName("sumOfDouble")\npublic fun Array<out
Double>.sum(): Double {\n  var sum: Double = 0.0\n  for (element in this) {\n    sum += element\n  }\n
return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*/\npublic fun ByteArray.sum(): Int {\n  var sum: Int = 0\n  for (element in this) {\n    sum += element\n
}\n  return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*/\npublic fun ShortArray.sum(): Int {\n  var sum: Int = 0\n  for (element in

```

```

this) {\n    sum += element\n } \n    return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n\n *\n\npublic fun IntArray.sum(): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n\n *\n\npublic fun LongArray.sum(): Long {\n    var sum: Long = 0L\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n\n *\n\npublic fun FloatArray.sum(): Float {\n    var sum: Float = 0.0f\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n\n *\n\npublic fun DoubleArray.sum(): Double {\n    var sum: Double = 0.0\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n"/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n\n *\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("RangesKt")\n\npackage\nkotlin.ranges\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.*\n\n/**\n * Returns the first element.\n\n *\n * @throws NoSuchElementException if the progression is empty.\n\n *\n\n@SinceKotlin("1.7")\npublic fun IntProgression.first(): Int {\n    if (isEmpty())\n        throw\n        NoSuchElementException("Progression $this is empty.")\n    return this.first\n}\n\n/**\n * Returns the first element.\n\n *\n * @throws NoSuchElementException if the progression is empty.\n\n *\n\n@SinceKotlin("1.7")\npublic fun LongProgression.first(): Long {\n    if (isEmpty())\n        throw\n        NoSuchElementException("Progression $this is empty.")\n    return this.first\n}\n\n/**\n * Returns the first element.\n\n *\n * @throws NoSuchElementException if the progression is empty.\n\n *\n\n@SinceKotlin("1.7")\npublic fun CharProgression.first(): Char {\n    if (isEmpty())\n        throw\n        NoSuchElementException("Progression $this is empty.")\n    return this.first\n}\n\n/**\n * Returns the first element, or `null` if the progression is empty.\n\n *\n\n@SinceKotlin("1.7")\npublic fun IntProgression.firstOrNull(): Int? {\n    return if (isEmpty()) null else this.first\n}\n\n/**\n * Returns the first element, or `null` if the progression is empty.\n\n *\n\n@SinceKotlin("1.7")\npublic fun LongProgression.firstOrNull(): Long? {\n    return if (isEmpty())\n        null else this.first\n}\n\n/**\n * Returns the first element, or `null` if the progression is empty.\n\n *\n\n@SinceKotlin("1.7")\npublic fun CharProgression.firstOrNull(): Char? {\n    return if (isEmpty())\n        null else this.first\n}\n\n/**\n * Returns the last element.\n\n *\n * @throws NoSuchElementException if the progression is empty.\n\n *\n * @sample samples.collections.Collections.Elements.last\n\n *\n\n@SinceKotlin("1.7")\npublic fun\nIntProgression.last(): Int {\n    if (isEmpty())\n        throw NoSuchElementException("Progression $this is\n        empty.")\n    return this.last\n}\n\n/**\n * Returns the last element.\n\n *\n * @throws NoSuchElementException if\n        the progression is empty.\n\n *\n * @sample samples.collections.Collections.Elements.last\n\n *\n\n@SinceKotlin("1.7")\npublic fun LongProgression.last(): Long {\n    if (isEmpty())\n        throw\n        NoSuchElementException("Progression $this is empty.")\n    return this.last\n}\n\n/**\n * Returns the last\n        element.\n\n *\n * @throws NoSuchElementException if the progression is empty.\n\n *\n * @sample\n        samples.collections.Collections.Elements.last\n\n *\n\n@SinceKotlin("1.7")\npublic fun CharProgression.last(): Char {\n    if (isEmpty())\n        throw\n        NoSuchElementException("Progression $this is empty.")\n    return\n        this.last\n}\n\n/**\n * Returns the last element, or `null` if the progression is empty.\n\n *\n * @sample\n        samples.collections.Collections.Elements.last\n\n *\n\n@SinceKotlin("1.7")\npublic fun IntProgression.lastOrNull():\nInt? {\n    return if (isEmpty()) null else this.last\n}\n\n/**\n * Returns the last element, or `null` if the progression is\n        empty.\n\n *\n * @sample\n        samples.collections.Collections.Elements.last\n\n *\n\n@SinceKotlin("1.7")\npublic fun LongProgression.lastOrNull(): Long? {\n    return if (isEmpty())\n        null else this.last\n}\n\n/**\n * Returns the last\n        element, or `null` if the progression is empty.\n\n *\n * @sample\n        samples.collections.Collections.Elements.last\n\n *\n\n@SinceKotlin("1.7")\npublic fun CharProgression.lastOrNull(): Char? {\n    return if (isEmpty())\n        null else\n        this.last\n}\n\n/**\n * Returns a random element from this range.\n\n *\n * @throws IllegalArgumentException if this\n        range is empty.\n\n *\n\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun IntRange.random(): Int {\n    return random(Random)\n}\n\n/**\n * Returns a random element from this range.\n\n *\n * @throws\n        IllegalArgumentException if this range is empty.\n\n *\n\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic

```

```

inline fun LongRange.random(): Long {\n    return random(Random)\n}\n\n/**\n * Returns a random element from
this range.\n * \n * @throws IllegalArgumentException if this range is empty.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun CharRange.random(): Char {\n    return
random(Random)\n}\n\n/**\n * Returns a random element from this range using the specified source of
randomness.\n * \n * @throws IllegalArgumentException if this range is empty.\n
*\n@SinceKotlin("1.3")\npublic fun IntRange.random(random: Random): Int {\n    try {\n        return
random.nextInt(this)\n    } catch(e: IllegalArgumentException) {\n        throw
NoSuchElementException(e.message)\n    }\n}\n\n/**\n * Returns a random element from this range using the
specified source of randomness.\n * \n * @throws IllegalArgumentException if this range is empty.\n
*\n@SinceKotlin("1.3")\npublic fun LongRange.random(random: Random): Long {\n    try {\n        return
random.nextLong(this)\n    } catch(e: IllegalArgumentException) {\n        throw
NoSuchElementException(e.message)\n    }\n}\n\n/**\n * Returns a random element from this range using the
specified source of randomness.\n * \n * @throws IllegalArgumentException if this range is empty.\n
*\n@SinceKotlin("1.3")\npublic fun CharRange.random(random: Random): Char {\n    try {\n        return
random.nextInt(first.code, last.code + 1).toChar()\n    } catch(e: IllegalArgumentException) {\n        throw
NoSuchElementException(e.message)\n    }\n}\n\n/**\n * Returns a random element from this range, or `null` if this
range is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun IntRange.randomOrNull(): Int? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a random
element from this range, or `null` if this range is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun LongRange.randomOrNull(): Long? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this range, or `null` if this range is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun CharRange.randomOrNull(): Char? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this range using the specified source of randomness, or `null` if this range is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
IntRange.randomOrNull(random: Random): Int? {\n    if (isEmpty())\n        return null\n    return
random.nextInt(this)\n}\n\n/**\n * Returns a random element from this range using the specified source of
randomness, or `null` if this range is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
LongRange.randomOrNull(random: Random): Long? {\n    if (isEmpty())\n        return null\n    return
random.nextLong(this)\n}\n\n/**\n * Returns a random element from this range using the specified source of
randomness, or `null` if this range is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
CharRange.randomOrNull(random: Random): Char? {\n    if (isEmpty())\n        return null\n    return
random.nextInt(first.code, last.code + 1).toChar()\n}\n\n/**\n * Returns `true` if this range contains the specified
[element].\n * \n * Always returns `false` if the [element] is `null`.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline operator fun IntRange.contains(element:
Int?): Boolean {\n    return element != null && contains(element)\n}\n\n/**\n * Returns `true` if this range contains
the specified [element].\n * \n * Always returns `false` if the [element] is `null`.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline operator fun LongRange.contains(element:
Long?): Boolean {\n    return element != null && contains(element)\n}\n\n/**\n * Returns `true` if this range
contains the specified [element].\n * \n * Always returns `false` if the [element] is `null`.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline operator fun CharRange.contains(element:
Char?): Boolean {\n    return element != null && contains(element)\n}\n\n/**\n * Checks if the specified [value]
belongs to this range.\n *\n@kotlin.jvm.JvmName("intRangeContains")\npublic operator fun
ClosedRange<Int>.contains(value: Byte): Boolean {\n    return contains(value.toInt())\n}\n\n/**\n * Checks if the

```

specified [value] belongs to this range.
`*\n@kotlin.jvm.JvmName("longRangeContains")\npublic operator fun ClosedRange<Long>.contains(value: Byte): Boolean {\n return contains(value.toLong())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName("shortRangeContains")\npublic operator fun ClosedRange<Short>.contains(value: Byte): Boolean {\n return contains(value.toShort())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince = "1.5")\n@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun ClosedRange<Double>.contains(value: Byte): Boolean {\n return contains(value.toDouble())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince = "1.5")\n@kotlin.jvm.JvmName("floatRangeContains")\npublic operator fun ClosedRange<Float>.contains(value: Byte): Boolean {\n return contains(value.toFloat())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName("intRangeContains")\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Int>.contains(value: Byte): Boolean {\n return contains(value.toInt())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName("longRangeContains")\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Long>.contains(value: Byte): Boolean {\n return contains(value.toLong())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName("shortRangeContains")\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Short>.contains(value: Byte): Boolean {\n return contains(value.toShort())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun IntRange.contains(value: Byte): Boolean {\n return (this as ClosedRange<Int>).contains(value)\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun LongRange.contains(value: Byte): Boolean {\n return (this as ClosedRange<Long>).contains(value)\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince = "1.5")\n@kotlin.jvm.JvmName("intRangeContains")\npublic operator fun ClosedRange<Int>.contains(value: Double): Boolean {\n return value.toIntExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince = "1.5")\n@kotlin.jvm.JvmName("longRangeContains")\npublic operator fun ClosedRange<Long>.contains(value: Double): Boolean {\n return value.toLongExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince = "1.5")\n@kotlin.jvm.JvmName("byteRangeContains")\npublic operator fun ClosedRange<Byte>.contains(value: Double): Boolean {\n return value.toByteExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince = "1.5")\n@kotlin.jvm.JvmName("shortRangeContains")\npublic operator fun ClosedRange<Short>.contains(value: Double): Boolean {\n return value.toShortExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */`

```

*\n@kotlin.jvm.JvmName("floatRangeContains")\npublic operator fun ClosedRange<Float>.contains(value:
Double): Boolean {\n    return contains(value.toFloat())\n}\n\n/**\n * Checks if the specified [value] belongs to this
range.\n */\n@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous
semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4",
hiddenSince = "1.5")\n@kotlin.jvm.JvmName("intRangeContains")\npublic operator fun
ClosedRange<Int>.contains(value: Float): Boolean {\n    return value.toIntExactOrNull().let { if (it != null)
contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics
and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince
= "1.5")\n@kotlin.jvm.JvmName("longRangeContains")\npublic operator fun
ClosedRange<Long>.contains(value: Float): Boolean {\n    return value.toLongExactOrNull().let { if (it != null)
contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics
and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince
= "1.5")\n@kotlin.jvm.JvmName("byteRangeContains")\npublic operator fun
ClosedRange<Byte>.contains(value: Float): Boolean {\n    return value.toByteExactOrNull().let { if (it != null)
contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics
and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince
= "1.5")\n@kotlin.jvm.JvmName("shortRangeContains")\npublic operator fun
ClosedRange<Short>.contains(value: Float): Boolean {\n    return value.toShortExactOrNull().let { if (it != null)
contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun ClosedRange<Double>.contains(value:
Float): Boolean {\n    return contains(value.toDouble())\n}\n\n/**\n * Checks if the specified [value] belongs to this
range.\n */\n@kotlin.jvm.JvmName("doubleRangeContains")\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\npublic
operator fun OpenEndRange<Double>.contains(value: Float): Boolean {\n    return
contains(value.toDouble())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName("longRangeContains")\npublic operator fun ClosedRange<Long>.contains(value: Int):
Boolean {\n    return contains(value.toLong())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName("byteRangeContains")\npublic operator fun ClosedRange<Byte>.contains(value: Int):
Boolean {\n    return value.toByteExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if
the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName("shortRangeContains")\npublic operator
fun ClosedRange<Short>.contains(value: Int): Boolean {\n    return value.toShortExactOrNull().let { if (it != null)
contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics
and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince
= "1.5")\n@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun
ClosedRange<Double>.contains(value: Int): Boolean {\n    return contains(value.toDouble())\n}\n\n/**\n * Checks
if the specified [value] belongs to this range.\n */\n@Deprecated("This `contains` operation mixing integer and
floating point arguments has ambiguous semantics and is going to be
removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince =
"1.5")\n@kotlin.jvm.JvmName("floatRangeContains")\npublic operator fun ClosedRange<Float>.contains(value:
Int): Boolean {\n    return contains(value.toFloat())\n}\n\n/**\n * Checks if the specified [value] belongs to this
range.\n */\n@kotlin.jvm.JvmName("longRangeContains")\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\npublic
operator fun OpenEndRange<Long>.contains(value: Int): Boolean {\n    return contains(value.toLong())\n}\n\n/**\n *
Checks if the specified [value] belongs to this range.\n */

```



```

*\n@kotlin.jvm.JvmName("byteRangeContains")\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\npublic
operator fun OpenEndRange<Byte>.contains(value: Int): Boolean {\n    return value.toByteExactOrNull().let { if (it
!= null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*\n@kotlin.jvm.JvmName("shortRangeContains")\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\npublic
operator fun OpenEndRange<Short>.contains(value: Int): Boolean {\n    return value.toShortExactOrNull().let { if
(it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun LongRange.contains(value: Int): Boolean {\n    return
(this as ClosedRange<Long>).contains(value)\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*\n@kotlin.jvm.JvmName("intRangeContains")\npublic operator fun ClosedRange<Int>.contains(value: Long):
Boolean {\n    return value.toIntExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the
specified [value] belongs to this range.\n
*\n@kotlin.jvm.JvmName("byteRangeContains")\npublic operator fun
ClosedRange<Byte>.contains(value: Long): Boolean {\n    return value.toByteExactOrNull().let { if (it != null)
contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*\n@kotlin.jvm.JvmName("shortRangeContains")\npublic operator fun ClosedRange<Short>.contains(value:
Long): Boolean {\n    return value.toShortExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n *
Checks if the specified [value] belongs to this range.\n
*\n@Deprecated("This `contains` operation mixing integer
and floating point arguments has ambiguous semantics and is going to be
removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince =
"1.5")\n@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun
ClosedRange<Double>.contains(value: Long): Boolean {\n    return contains(value.toDouble())\n}\n\n/**\n *
Checks if the specified [value] belongs to this range.\n
*\n@Deprecated("This `contains` operation mixing integer
and floating point arguments has ambiguous semantics and is going to be
removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince =
"1.5")\n@kotlin.jvm.JvmName("floatRangeContains")\npublic operator fun ClosedRange<Float>.contains(value:
Long): Boolean {\n    return contains(value.toFloat())\n}\n\n/**\n * Checks if the specified [value] belongs to this
range.\n
*\n@kotlin.jvm.JvmName("intRangeContains")\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\npublic
operator fun OpenEndRange<Int>.contains(value: Long): Boolean {\n    return value.toIntExactOrNull().let { if (it
!= null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*\n@kotlin.jvm.JvmName("byteRangeContains")\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\npublic
operator fun OpenEndRange<Byte>.contains(value: Long): Boolean {\n    return value.toByteExactOrNull().let { if
(it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*\n@kotlin.jvm.JvmName("shortRangeContains")\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\npublic
operator fun OpenEndRange<Short>.contains(value: Long): Boolean {\n    return value.toShortExactOrNull().let {
if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun IntRange.contains(value: Long): Boolean {\n    return
(this as ClosedRange<Int>).contains(value)\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*\n@kotlin.jvm.JvmName("intRangeContains")\npublic operator fun ClosedRange<Int>.contains(value: Short):
Boolean {\n    return contains(value.toInt())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*\n@kotlin.jvm.JvmName("longRangeContains")\npublic operator fun ClosedRange<Long>.contains(value:
Short): Boolean {\n    return contains(value.toLong())\n}\n\n/**\n * Checks if the specified [value] belongs to this
range.\n
*\n@kotlin.jvm.JvmName("byteRangeContains")\npublic operator fun
ClosedRange<Byte>.contains(value: Short): Boolean {\n    return value.toByteExactOrNull().let { if (it != null)
contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*\n@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics
and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince
= "1.5")\n@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun
ClosedRange<Double>.contains(value: Short): Boolean {\n    return contains(value.toDouble())\n}\n\n/**\n *

```

Checks if the specified [value] belongs to this range.
`@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")`
`@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4", hiddenSince = "1.5")`
`@kotlin.jvm.JvmName("floatRangeContains")`
`public operator fun ClosedRange<Float>.contains(value: Short): Boolean {`
`return contains(value.toFloat())`
`}`
`* Checks if the specified [value] belongs to this range.`

`@kotlin.jvm.JvmName("intRangeContains")`
`@SinceKotlin("1.7")`
`@ExperimentalStdlibApi`
`public operator fun OpenEndRange<Int>.contains(value: Short): Boolean {`
`return contains(value.toInt())`
`}`
`* Checks if the specified [value] belongs to this range.`

`@kotlin.jvm.JvmName("longRangeContains")`
`@SinceKotlin("1.7")`
`@ExperimentalStdlibApi`
`public operator fun OpenEndRange<Long>.contains(value: Short): Boolean {`
`return contains(value.toLong())`
`}`
`* Checks if the specified [value] belongs to this range.`

`@kotlin.jvm.JvmName("byteRangeContains")`
`@SinceKotlin("1.7")`
`@ExperimentalStdlibApi`
`public operator fun OpenEndRange<Byte>.contains(value: Short): Boolean {`
`return value.toByteExactOrNull().let { if (it != null) contains(it) else false }`
`}`
`* Checks if the specified [value] belongs to this range.`

`@kotlin.internal.InlineOnly`
`public inline operator fun IntRange.contains(value: Short): Boolean {`
`return (this as ClosedRange<Int>).contains(value)`
`}`
`* Checks if the specified [value] belongs to this range.`

`@kotlin.internal.InlineOnly`
`public inline operator fun LongRange.contains(value: Short): Boolean {`
`return (this as ClosedRange<Long>).contains(value)`
`}`
`* Returns a progression from this value down to the specified [to] value with the step -1.`
`* The [to] value should be less than or equal to `this` value.`
`* If the [to] value is greater than `this` value the returned progression is empty.`
`public infix fun Int.downTo(to: Byte): IntProgression {`
`return IntProgression.fromClosedRange(this, to.toInt(), -1)`
`}`
`* Returns a progression from this value down to the specified [to] value with the step -1.`
`* The [to] value should be less than or equal to `this` value.`
`* If the [to] value is greater than `this` value the returned progression is empty.`
`public infix fun Long.downTo(to: Byte): LongProgression {`
`return LongProgression.fromClosedRange(this, to.toLong(), -1L)`
`}`
`* Returns a progression from this value down to the specified [to] value with the step -1.`
`* The [to] value should be less than or equal to `this` value.`
`* If the [to] value is greater than `this` value the returned progression is empty.`
`public infix fun Byte.downTo(to: Byte): IntProgression {`
`return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)`
`}`
`* Returns a progression from this value down to the specified [to] value with the step -1.`
`* The [to] value should be less than or equal to `this` value.`
`* If the [to] value is greater than `this` value the returned progression is empty.`
`public infix fun Short.downTo(to: Byte): IntProgression {`
`return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)`
`}`
`* Returns a progression from this value down to the specified [to] value with the step -1.`
`* The [to] value should be less than or equal to `this` value.`
`* If the [to] value is greater than `this` value the returned progression is empty.`
`public infix fun Char.downTo(to: Char): CharProgression {`
`return CharProgression.fromClosedRange(this, to, -1)`
`}`
`* Returns a progression from this value down to the specified [to] value with the step -1.`
`* The [to] value should be less than or equal to `this` value.`
`* If the [to] value is greater than `this` value the returned progression is empty.`
`public infix fun Int.downTo(to: Int): IntProgression {`
`return IntProgression.fromClosedRange(this, to, -1)`
`}`
`* Returns a progression from this value down to the specified [to] value with the step -1.`
`* The [to] value should be less than or equal to `this` value.`
`* If the [to] value is greater than `this` value the returned progression is empty.`
`public infix fun Long.downTo(to: Int): LongProgression {`
`return LongProgression.fromClosedRange(this, to.toLong(), -1L)`
`}`
`* Returns a progression from this value down to the specified [to] value with the step -1.`
`* The [to] value should be less than or equal to `this` value.`
`* If the [to] value is greater than `this` value the returned progression is empty.`
`public infix fun Byte.downTo(to: Int): IntProgression {`
`return IntProgression.fromClosedRange(this.toInt(), to, -1)`
`}`
`* Returns a progression from this value down to the specified [to] value with the step -1.`
`* The [to] value should be less than or equal to `this` value.`
`* If the [to] value is greater than `this` value the returned progression is empty.`
`public infix fun Short.downTo(to: Int):`

```

IntProgression {\n    return IntProgression.fromClosedRange(this.toInt(), to, -1)\n}\n\n/**\n * Returns a progression
from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal
to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\npublic infix
fun Int.downTo(to: Long): LongProgression {\n    return LongProgression.fromClosedRange(this.toLong(), to, -
1L)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n *
The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the
returned progression is empty.\n */\npublic infix fun Long.downTo(to: Long): LongProgression {\n    return
LongProgression.fromClosedRange(this, to, -1L)\n}\n\n/**\n * Returns a progression from this value down to the
specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to]
value is greater than `this` value the returned progression is empty.\n */\npublic infix fun Byte.downTo(to: Long):
LongProgression {\n    return LongProgression.fromClosedRange(this.toLong(), to, -1L)\n}\n\n/**\n * Returns a
progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less
than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n
*/\npublic infix fun Short.downTo(to: Long): LongProgression {\n    return
LongProgression.fromClosedRange(this.toLong(), to, -1L)\n}\n\n/**\n * Returns a progression from this value
down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n
*/\npublic infix fun Int.downTo(to:
Short): IntProgression {\n    return IntProgression.fromClosedRange(this, to.toInt(), -1)\n}\n\n/**\n * Returns a
progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less
than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n
*/\npublic infix fun Long.downTo(to: Short): LongProgression {\n    return LongProgression.fromClosedRange(this,
to.toLong(), -1L)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -
1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this`
value the returned progression is empty.\n */\npublic infix fun Byte.downTo(to: Short): IntProgression {\n    return
IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)\n}\n\n/**\n * Returns a progression from this value
down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n
*/\npublic infix fun
Short.downTo(to: Short): IntProgression {\n    return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -
1)\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value
is less than or equal to `this` value, then the returned range is empty.\n */\n\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline operator fun
Int.rangeUntil(to: Byte): IntRange {\n    return until(to)\n}\n\n/**\n * Returns a range from this value up to but
excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned
range is empty.\n */\n\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline
operator fun Long.rangeUntil(to: Byte): LongRange {\n    return until(to)\n}\n\n/**\n * Returns a range from this
value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value,
then the returned range is empty.\n */\n\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline operator fun
Byte.rangeUntil(to: Byte): IntRange {\n    return until(to)\n}\n\n/**\n * Returns a range from this value up to but
excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned
range is empty.\n */\n\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline
operator fun Short.rangeUntil(to: Byte): IntRange {\n    return until(to)\n}\n\n/**\n * Returns a range from this
value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value,
then the returned range is empty.\n */\n\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline operator fun
Char.rangeUntil(to: Char): CharRange {\n    return until(to)\n}\n\n/**\n * Returns a range from this value up to but
excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned
range is empty.\n */\n\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline

```

operator fun Int.rangeUntil(to: Int): IntRange { \n return until(to)\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n

```
*\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline operator fun Long.rangeUntil(to: Int): LongRange { \n return until(to)\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n *\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline operator fun Byte.rangeUntil(to: Int): IntRange { \n return until(to)\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n
```

```
*\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline operator fun Short.rangeUntil(to: Int): IntRange { \n return until(to)\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n *\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline operator fun Int.rangeUntil(to: Long): LongRange { \n return until(to)\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n
```

```
*\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline operator fun Long.rangeUntil(to: Long): LongRange { \n return until(to)\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n *\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline operator fun Byte.rangeUntil(to: Long): LongRange { \n return until(to)\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n
```

```
*\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline operator fun Short.rangeUntil(to: Long): LongRange { \n return until(to)\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n *\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline operator fun Int.rangeUntil(to: Short): IntRange { \n return until(to)\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n
```

```
*\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline operator fun Long.rangeUntil(to: Short): LongRange { \n return until(to)\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n *\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline operator fun Byte.rangeUntil(to: Short): IntRange { \n return until(to)\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n
```

```
*\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline operator fun Short.rangeUntil(to: Short): IntRange { \n return until(to)\n}\n\n/**\n * Returns a progression that goes over the same range in the opposite direction with the same step.\n *\npublic fun IntProgression.reversed(): IntProgression { \n return IntProgression.fromClosedRange(last, first, -step)\n}\n\n/**\n * Returns a progression that goes over the same range in the opposite direction with the same step.\n *\npublic fun LongProgression.reversed():
```

```
LongProgression { \n return LongProgression.fromClosedRange(last, first, -step)\n}\n\n/**\n * Returns a progression that goes over the same range in the opposite direction with the same step.\n *\npublic fun CharProgression.reversed(): CharProgression { \n return CharProgression.fromClosedRange(last, first, -step)\n}\n\n/**\n * Returns a progression that goes over the same range with the given step.\n *\npublic infix fun IntProgression.step(step: Int): IntProgression { \n checkStepIsPositive(step > 0, step)\n return
```



```

1).toLong()\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the
[to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Byte.until(to:
Int): IntRange {\n    if (to <= Int.MIN_VALUE) return IntRange.EMPTY\n    return this.toInt() .. (to -
1).toInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the
[to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Short.until(to:
Int): IntRange {\n    if (to <= Int.MIN_VALUE) return IntRange.EMPTY\n    return this.toInt() .. (to -
1).toInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the
[to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Int.until(to:
Long): LongRange {\n    if (to <= Long.MIN_VALUE) return LongRange.EMPTY\n    return this.toLong() .. (to -
1).toLong()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the
[to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Long.until(to:
Long): LongRange {\n    if (to <= Long.MIN_VALUE) return LongRange.EMPTY\n    return this .. (to -
1).toLong()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the
[to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Byte.until(to:
Long): LongRange {\n    if (to <= Long.MIN_VALUE) return LongRange.EMPTY\n    return this.toLong() .. (to -
1).toLong()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the
[to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Short.until(to:
Long): LongRange {\n    if (to <= Long.MIN_VALUE) return LongRange.EMPTY\n    return this.toLong() .. (to -
1).toLong()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the
[to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Int.until(to:
Short): IntRange {\n    return this .. (to.toInt() - 1).toInt()\n}\n\n/**\n * Returns a range from this value up to but
excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned
range is empty.\n */\npublic infix fun Long.until(to: Short): LongRange {\n    return this .. (to.toLong() -
1).toLong()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the
[to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Byte.until(to:
Short): IntRange {\n    return this.toInt() .. (to.toInt() - 1).toInt()\n}\n\n/**\n * Returns a range from this value up to
but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned
range is empty.\n */\npublic infix fun Short.until(to: Short): IntRange {\n    return this.toInt() .. (to.toInt() -
1).toInt()\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this
value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtLeastComparable\n */\npublic fun <T : Comparable<T>>
T.coerceAtLeast(minimumValue: T): T {\n    return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater
than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun Byte.coerceAtLeast(minimumValue: Byte):
Byte {\n    return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less
than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue]
or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun Short.coerceAtLeast(minimumValue: Short): Short {\n    return if (this < minimumValue)
minimumValue else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n *
@return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n *
@sample samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun Int.coerceAtLeast(minimumValue:
Int): Int {\n    return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not
less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the
[minimumValue] or the [minimumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun Long.coerceAtLeast(minimumValue: Long):
Long {\n    return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less
than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue]

```

```

or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeast\n
*/\npublic fun Float.coerceAtLeast(minimumValue: Float): Float {\n    return if (this < minimumValue)\n    minimumValue else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun\nDouble.coerceAtLeast(minimumValue: Double): Double {\n    return if (this < minimumValue) minimumValue else\n    this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this\n    value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample\n    samples.comparisons.ComparableOps.coerceAtMostComparable\n */\npublic fun <T : Comparable<T>>\nT.coerceAtMost(maximumValue: T): T {\n    return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this value if it's less\n    than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample\n    samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun Byte.coerceAtMost(maximumValue: Byte):\nByte {\n    return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not\n    greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the\n    [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample\n    samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun Short.coerceAtMost(maximumValue: Short):\nShort {\n    return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not\n    greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the\n    [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample\n    samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun Int.coerceAtMost(maximumValue: Int): Int\n{\n    return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater\n    than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the [maximumValue] or\n    the [maximumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun Long.coerceAtMost(maximumValue: Long): Long {\n    return if (this > maximumValue)\n    maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun\nFloat.coerceAtMost(maximumValue: Float): Float {\n    return if (this > maximumValue) maximumValue else\n    this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this\n    value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample\n    samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun Double.coerceAtMost(maximumValue:\nDouble): Double {\n    return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this\n    value lies in the specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range,\n    or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than\n    [maximumValue].\n * \n * @sample samples.comparisons.ComparableOps.coerceInComparable\n */\npublic fun\n<T : Comparable<T>> T.coerceIn(minimumValue: T?, maximumValue: T?): T {\n    if (minimumValue !== null\n    && maximumValue !== null) {\n        if (minimumValue > maximumValue) throw\n        IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than\n        minimum $minimumValue.")\n        if (this < minimumValue) return minimumValue\n        if (this >\n        maximumValue) return maximumValue\n    }\n    else {\n        if (minimumValue !== null && this <\n        minimumValue) return minimumValue\n        if (maximumValue !== null && this > maximumValue) return\n        maximumValue\n    }\n    return this\n}\n\n/**\n * Ensures that this value lies in the specified range\n    [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value\n    is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n * @sample\n    samples.comparisons.ComparableOps.coerceIn\n */\npublic fun Byte.coerceIn(minimumValue: Byte,\nmaximumValue: Byte): Byte {\n    if (minimumValue > maximumValue) throw\n    IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than

```

```

minimum $minimumValue.)\n if (this < minimumValue) return minimumValue\n if (this > maximumValue)
return maximumValue\n return this\n}\n\n/**\n * Ensures that this value lies in the specified range
[minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value
is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n * @sample
samples.comparisons.ComparableOps.coerceIn\n *^\npublic fun Short.coerceIn(minimumValue: Short,
maximumValue: Short): Short {\n if (minimumValue > maximumValue) throw
IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than
minimum $minimumValue.")\n if (this < minimumValue) return minimumValue\n if (this > maximumValue)
return maximumValue\n return this\n}\n\n/**\n * Ensures that this value lies in the specified range
[minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value
is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n * @sample
samples.comparisons.ComparableOps.coerceIn\n *^\npublic fun Int.coerceIn(minimumValue: Int, maximumValue:
Int): Int {\n if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an
empty range: maximum $maximumValue is less than minimum $minimumValue.")\n if (this < minimumValue)
return minimumValue\n if (this > maximumValue) return maximumValue\n return this\n}\n\n/**\n * Ensures
that this value lies in the specified range [minimumValue]..[maximumValue].\n * \n * @return this value in
the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater
than [maximumValue].\n * \n * @sample samples.comparisons.ComparableOps.coerceIn\n *^\npublic fun
Long.coerceIn(minimumValue: Long, maximumValue: Long): Long {\n if (minimumValue > maximumValue)
throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than
minimum $minimumValue.")\n if (this < minimumValue) return minimumValue\n if (this > maximumValue)
return maximumValue\n return this\n}\n\n/**\n * Ensures that this value lies in the specified range
[minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value
is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n * @sample
samples.comparisons.ComparableOps.coerceIn\n *^\npublic fun Float.coerceIn(minimumValue: Float,
maximumValue: Float): Float {\n if (minimumValue > maximumValue) throw
IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than
minimum $minimumValue.")\n if (this < minimumValue) return minimumValue\n if (this > maximumValue)
return maximumValue\n return this\n}\n\n/**\n * Ensures that this value lies in the specified range
[minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value
is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n * @sample
samples.comparisons.ComparableOps.coerceIn\n *^\npublic fun Double.coerceIn(minimumValue: Double,
maximumValue: Double): Double {\n if (minimumValue > maximumValue) throw
IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than
minimum $minimumValue.")\n if (this < minimumValue) return minimumValue\n if (this > maximumValue)
return maximumValue\n return this\n}\n\n/**\n * Ensures that this value lies in the specified [range].\n * \n *
@return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or `range.endInclusive`
if this value is greater than `range.endInclusive`.\n * \n * @sample
samples.comparisons.ComparableOps.coerceInFloatingPointRange\n *^\n@SinceKotlin("1.1")\npublic fun <T :
Comparable<T>> T.coerceIn(range: ClosedFloatingPointRange<T>): T {\n if (range.isEmpty()) throw
IllegalArgumentException("Cannot coerce value to an empty range: $range.")\n return when {\n // this <
start equiv to this <= start && !(this >= start)\n range.lessThanOrEqualTo(this, range.start) &&
!range.lessThanOrEqualTo(range.start, this) -> range.start\n // this > end equiv to this >= end && !(this <= end)\n
range.lessThanOrEqualTo(range.endInclusive, this) && !range.lessThanOrEqualTo(this, range.endInclusive) ->
range.endInclusive\n else -> this\n } }\n\n/**\n * Ensures that this value lies in the specified [range].\n * \n *
@return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or `range.endInclusive`
if this value is greater than `range.endInclusive`.\n * \n * @sample
samples.comparisons.ComparableOps.coerceInComparable\n *^\npublic fun <T : Comparable<T>>

```



```

T.coerceIn(range: ClosedRange<T>): T {
    if (range is ClosedFloatingPointRange) {
        return this.coerceIn<T>(range)
    }
    if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to an empty range: $range.")
    return when {
        this < range.start -> range.start
        this > range.endInclusive -> range.endInclusive
        else -> this
    }
}

/** Ensures that this value lies in the specified [range].
 * @return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`.
 * @sample samples.comparisons.ComparableOps.coerceIn
 */
public fun Int.coerceIn(range: ClosedRange<Int>): Int {
    if (range is ClosedFloatingPointRange) {
        return this.coerceIn<Int>(range)
    }
    if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to an empty range: $range.")
    return when {
        this < range.start -> range.start
        this > range.endInclusive -> range.endInclusive
        else -> this
    }
}

/** Ensures that this value lies in the specified [range].
 * @return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`.
 * @sample samples.comparisons.ComparableOps.coerceIn
 */
public fun Long.coerceIn(range: ClosedRange<Long>): Long {
    if (range is ClosedFloatingPointRange) {
        return this.coerceIn<Long>(range)
    }
    if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to an empty range: $range.")
    return when {
        this < range.start -> range.start
        this > range.endInclusive -> range.endInclusive
        else -> this
    }
}

/** Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
// Auto-generated file. DO NOT EDIT!
package kotlin.nimpor
import kotlin.experimental.*
import kotlin.jvm.*
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
@JvmInline
public value class UByte @kotlin.internal.IntrinsicConstEvaluation @PublishedApi internal constructor(@PublishedApi internal val data: Byte) : Comparable<UByte> {
    companion object {
        /** A constant holding the minimum value an instance of UByte can have.
         * public const val MIN_VALUE: UByte = UByte(0)
         * A constant holding the maximum value an instance of UByte can have.
         * public const val MAX_VALUE: UByte = UByte(-1)
         * The number of bytes used to represent an instance of UByte in a binary form.
         * public const val SIZE_BYTES: Int = 1
         * The number of bits used to represent an instance of UByte in a binary form.
         * public const val SIZE_BITS: Int = 8
         * Compares this value with the specified value for order.
         * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,
         * or a positive number if it's greater than other.
         * @kotlin.internal.InlineOnly
         * @Suppress("OVERRIDE_BY_INLINE")
         * public override inline operator fun compareTo(other: UByte): Int = this.toInt().compareTo(other.toInt())
         * Compares this value with the specified value for order.
         * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,
         * or a positive number if it's greater than other.
         * @kotlin.internal.InlineOnly
         * public inline operator fun compareTo(other: UShort): Int = this.toInt().compareTo(other.toInt())
         * Compares this value with the specified value for order.
         * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,
         * or a positive number if it's greater than other.
         * @kotlin.internal.InlineOnly
         * public inline operator fun compareTo(other: UInt): Int = this.toUInt().compareTo(other)
         * Compares this value with the specified value for order.
         * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,
         * or a positive number if it's greater than other.
         * @kotlin.internal.InlineOnly
         * public inline operator fun compareTo(other: ULong): Int = this.toULong().compareTo(other)
         * Adds the other value to this value.
         * @kotlin.internal.InlineOnly
         * public inline operator fun plus(other: UByte): UInt = this.toUInt().plus(other.toUInt())
         * Adds the other value to this value.
         * @kotlin.internal.InlineOnly
         * public inline operator fun plus(other: UShort): UInt = this.toUInt().plus(other.toUInt())
         * Adds the other value to this value.
         * @kotlin.internal.InlineOnly
         * public inline operator fun plus(other: UInt): UInt = this.toUInt().plus(other)
         * Adds the other value to this value.
         * @kotlin.internal.InlineOnly
         * public inline operator fun plus(other: ULong): ULong = this.toULong().plus(other)
         * Subtracts the other value from this value.
         * @kotlin.internal.InlineOnly

```

```

public inline operator fun minus(other: UByte): UInt = this.toUInt().minus(other.toUInt())\n /** Subtracts the
other value from this value. *\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: UShort):
UInt = this.toUInt().minus(other.toUInt())\n /** Subtracts the other value from this value. *\n
@kotlin.internal.InlineOnly\n public inline operator fun minus(other: UInt): UInt = this.toUInt().minus(other)\n
/** Subtracts the other value from this value. *\n @kotlin.internal.InlineOnly\n public inline operator fun
minus(other: ULong): ULong = this.toULong().minus(other)\n\n /** Multiplies this value by the other value. *\n
@kotlin.internal.InlineOnly\n public inline operator fun times(other: UByte): UInt =
this.toUInt().times(other.toUInt())\n /** Multiplies this value by the other value. *\n
@kotlin.internal.InlineOnly\n public inline operator fun times(other: UShort): UInt =
this.toUInt().times(other.toUInt())\n /** Multiplies this value by the other value. *\n
@kotlin.internal.InlineOnly\n public inline operator fun times(other: UInt): UInt = this.toUInt().times(other)\n
/** Multiplies this value by the other value. *\n @kotlin.internal.InlineOnly\n public inline operator fun
times(other: ULong): ULong = this.toULong().times(other)\n\n /** Divides this value by the other value,
truncating the result to an integer that is closer to zero. *\n @kotlin.internal.InlineOnly\n public inline operator
fun div(other: UByte): UInt = this.toUInt().div(other.toUInt())\n /** Divides this value by the other value,
truncating the result to an integer that is closer to zero. *\n @kotlin.internal.InlineOnly\n public inline operator
fun div(other: UShort): UInt = this.toUInt().div(other.toUInt())\n /** Divides this value by the other value,
truncating the result to an integer that is closer to zero. *\n @kotlin.internal.InlineOnly\n public inline operator
fun div(other: UInt): UInt = this.toUInt().div(other)\n /** Divides this value by the other value, truncating the
result to an integer that is closer to zero. *\n @kotlin.internal.InlineOnly\n public inline operator fun div(other:
ULong): ULong = this.toULong().div(other)\n\n /**\n * Calculates the remainder of truncating division of this
value by the other value.\n * \n * The result is always less than the divisor.\n *\n
@kotlin.internal.InlineOnly\n public inline operator fun rem(other: UByte): UInt =
this.toUInt().rem(other.toUInt())\n /**\n * Calculates the remainder of truncating division of this value by the
other value.\n * \n * The result is always less than the divisor.\n *\n @kotlin.internal.InlineOnly\n public
inline operator fun rem(other: UShort): UInt = this.toUInt().rem(other.toUInt())\n /**\n * Calculates the
remainder of truncating division of this value by the other value.\n * \n * The result is always less than the
divisor.\n *\n @kotlin.internal.InlineOnly\n public inline operator fun rem(other: UInt): UInt =
this.toUInt().rem(other)\n /**\n * Calculates the remainder of truncating division of this value by the other
value.\n * \n * The result is always less than the divisor.\n *\n @kotlin.internal.InlineOnly\n public
inline operator fun rem(other: ULong): ULong = this.toULong().rem(other)\n\n /**\n * Divides this value by
the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned
types, the results of flooring division and truncating division are the same.\n *\n @kotlin.internal.InlineOnly\n
public inline fun floorDiv(other: UByte): UInt = this.toUInt().floorDiv(other.toUInt())\n /**\n * Divides this
value by the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned
types, the results of flooring division and truncating division are the same.\n *\n @kotlin.internal.InlineOnly\n
public inline fun floorDiv(other: UShort): UInt = this.toUInt().floorDiv(other.toUInt())\n /**\n * Divides this
value by the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned
types, the results of flooring division and truncating division are the same.\n *\n @kotlin.internal.InlineOnly\n
public inline fun floorDiv(other: UInt): UInt = this.toUInt().floorDiv(other)\n /**\n * Divides this value by the
other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned types,
the results of flooring division and truncating division are the same.\n *\n @kotlin.internal.InlineOnly\n public
inline fun floorDiv(other: ULong): ULong = this.toULong().floorDiv(other)\n\n /**\n * Calculates the
remainder of flooring division of this value by the other value.\n * \n * The result is always less than the
divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating division are the same.\n
*\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UByte): UByte =
this.toUInt().mod(other.toUInt()).toUByte()\n /**\n * Calculates the remainder of flooring division of this value
by the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the

```



```

@kotlin.internal.InlineOnly\n public inline fun toULong(): ULong = ULong(data.toLong() and 0xFF)\n\n /**\n * Converts this [UByte] value to [Float].\n * \n * The resulting `Float` value represents the same numerical\n value as this `UByte`.\n * \n @kotlin.internal.InlineOnly\n public inline fun toFloat(): Float =\n this.toInt().toFloat()\n /**\n * Converts this [UByte] value to [Double].\n * \n * The resulting `Double`\n value represents the same numerical value as this `UByte`.\n * \n @kotlin.internal.InlineOnly\n public inline\n fun toDouble(): Double = this.toInt().toDouble()\n\n public override fun toString(): String =\n toInt().toString()\n\n}\n\n**\n * Converts this [Byte] value to [UByte].\n * \n * If this value is positive, the resulting\n `UByte` value represents the same numerical value as this `Byte`.\n * \n * The resulting `UByte` value has the same\n binary representation as this `Byte` value.\n\n*\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n public inline fun Byte.toUByte(): UByte = UByte(this)\n /**\n * Converts this [Short] value to [UByte].\n * \n * If\n this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents\n * the same numerical value as this `Short`.\n * \n * The resulting `UByte` value is represented by the least significant 8\n bits of this `Short` value.\n\n*\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n public inline fun Short.toUByte(): UByte = UByte(this.toByte())\n /**\n * Converts this [Int] value to [UByte].\n * \n * If\n this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value\n represents\n * the same numerical value as this `Int`.\n * \n * The resulting `UByte` value is represented by the least\n significant 8 bits of this `Int` value.\n\n*\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n public inline fun Int.toUByte(): UByte = UByte(this.toByte())\n /**\n * Converts this [Long] value to [UByte].\n * \n * If\n this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value\n represents\n * the same numerical value as this `Long`.\n * \n * The resulting `UByte` value is represented by the\n least significant 8 bits of this `Long` value.\n\n*\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n public inline fun Long.toUByte(): UByte = UByte(this.toByte())\n", "/*\n * Copyright 2010-2022 JetBrains s.r.o.\n and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license\n that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage\nkotlin\n\nimport kotlin.experimental.*\nimport\nkotlin.jvm.*\n\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @JvmInline\n public value class UInt @kotlin.internal.IntrinsicConstEvaluation @PublishedApi internal constructor(@PublishedApi\n internal val data: Int) : Comparable<UInt> {\n\n companion object {\n\n /**\n * A constant holding the\n minimum value an instance of UInt can have.\n * \n public const val MIN_VALUE: UInt = UInt(0)\n\n /**\n * A constant holding the maximum value an instance of UInt can have.\n * \n public const val\n MAX_VALUE: UInt = UInt(-1)\n\n /**\n * The number of bytes used to represent an instance of UInt in a\n binary form.\n * \n public const val SIZE_BYTES: Int = 4\n\n /**\n * The number of bits used to\n represent an instance of UInt in a binary form.\n * \n public const val SIZE_BITS: Int = 32\n }\n\n /**\n * Compares this value with the specified value for order.\n * \n * Returns zero if this value is equal to the\n specified other value, a negative number if it's less than other,\n * or a positive number if it's greater than other.\n * \n *\n @kotlin.internal.InlineOnly\n public inline operator fun compareTo(other: UByte): Int =\n this.compareTo(other.toUInt())\n\n /**\n * Compares this value with the specified value for order.\n * \n * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n * or a\n positive number if it's greater than other.\n * \n *\n @kotlin.internal.InlineOnly\n public inline operator fun\n compareTo(other: UShort): Int = this.compareTo(other.toUInt())\n\n /**\n * Compares this value with the\n specified value for order.\n * \n * Returns zero if this value is equal to the specified other value, a negative number if\n it's less than other,\n * or a positive number if it's greater than other.\n * \n *\n @kotlin.internal.InlineOnly\n @Suppress("OVERRIDE_BY_INLINE")\n public override inline operator fun compareTo(other: UInt): Int =\n uintCompare(this.data, other.data)\n\n /**\n * Compares this value with the specified value for order.\n * \n

```

Returns zero if this value is equal to the specified other value, a negative number if it's less than other, or a positive number if it's greater than other.

```

@kotlin.internal.InlineOnly
public inline operator fun
compareTo(other: ULong): Int = this.toULong().compareTo(other)
/** Adds the other value to this value. */
@kotlin.internal.InlineOnly
public inline operator fun plus(other: UByte): UInt = this.plus(other.toUInt())
/** Adds the other value to this value. */
@kotlin.internal.InlineOnly
public inline operator fun plus(other: UShort): UInt = this.plus(other.toUInt())
/** Adds the other value to this value. */
@kotlin.internal.InlineOnly
public inline operator fun plus(other: UInt): UInt = UInt(this.data.plus(other.data))
/** Adds the other value to this value. */
@kotlin.internal.InlineOnly
public inline operator fun plus(other: ULong): ULong = this.toULong().plus(other)
/** Subtracts the other value from this value. */
@kotlin.internal.InlineOnly
public inline operator fun minus(other: UByte): UInt = this.minus(other.toUInt())
/** Subtracts the other value from this value. */
@kotlin.internal.InlineOnly
public inline operator fun minus(other: UShort): UInt = this.minus(other.toUInt())
/** Subtracts the other value from this value. */
@kotlin.internal.InlineOnly
public inline operator fun minus(other: UInt): UInt =
UInt(this.data.minus(other.data))
/** Subtracts the other value from this value. */
@kotlin.internal.InlineOnly
public inline operator fun minus(other: ULong): ULong =
this.toULong().minus(other)
/** Multiplies this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun times(other: UByte): UInt = this.times(other.toUInt())
/** Multiplies this value by the
other value. */
@kotlin.internal.InlineOnly
public inline operator fun times(other: UShort): UInt =
this.times(other.toUInt())
/** Multiplies this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun times(other: UInt): UInt = UInt(this.data.times(other.data))
/** Multiplies this value
by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun times(other: ULong): ULong =
this.toULong().times(other)
/** Divides this value by the other value, truncating the result to an integer that is
closer to zero. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UByte): UInt =
this.div(other.toUInt())
/** Divides this value by the other value, truncating the result to an integer that is closer
to zero. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UShort): UInt =
this.div(other.toUInt())
/** Divides this value by the other value, truncating the result to an integer that is closer
to zero. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UInt): UInt = uintDivide(this,
other)
/** Divides this value by the other value, truncating the result to an integer that is closer to zero. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: ULong): ULong =
this.toULong().div(other)
/**
 * Calculates the remainder of truncating division of this value by the other
value.
 * The result is always less than the divisor.
 */
@kotlin.internal.InlineOnly
public
inline operator fun rem(other: UByte): UInt = this.rem(other.toUInt())
/**
 * Calculates the remainder of
truncating division of this value by the other value.
 * The result is always less than the divisor.
 */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: UShort): UInt = this.rem(other.toUInt())
/**
 * Calculates the remainder of truncating division of this value by the other value.
 * The result is
always less than the divisor.
 */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: UInt):
UInt = uintRemainder(this, other)
/**
 * Calculates the remainder of truncating division of this value by the
other value.
 * The result is always less than the divisor.
 */
@kotlin.internal.InlineOnly
public
inline operator fun rem(other: ULong): ULong = this.toULong().rem(other)
/**
 * Divides this value by
the other value, flooring the result to an integer that is closer to negative infinity.
 * For unsigned types,
the results of flooring division and truncating division are the same.
 */
@kotlin.internal.InlineOnly
public
inline fun floorDiv(other: UByte): UInt = this.floorDiv(other.toUInt())
/**
 * Divides this value by the
other value, flooring the result to an integer that is closer to negative infinity.
 * For unsigned types, the
results of flooring division and truncating division are the same.
 */
@kotlin.internal.InlineOnly
public
inline fun floorDiv(other: UShort): UInt = this.floorDiv(other.toUInt())
/**
 * Divides this value by the other
value, flooring the result to an integer that is closer to negative infinity.
 * For unsigned types, the results
of flooring division and truncating division are the same.
 */
@kotlin.internal.InlineOnly
public inline
fun floorDiv(other: UInt): UInt = div(other)
/**
 * Divides this value by the other value, flooring the result to

```

```

an integer that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division and
truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other:
ULong): ULong = this.toULong().floorDiv(other)\n\n /**\n * Calculates the remainder of flooring division of
this value by the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned
types, the remainders of flooring division and truncating division are the same.\n */\n
@kotlin.internal.InlineOnly\n public inline fun mod(other: UByte): UByte = this.mod(other.toUInt()).toUByte()\n
/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is
always less than the divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating
division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UShort): UShort =
this.mod(other.toUInt()).toUShort()\n\n /**\n * Calculates the remainder of flooring division of this value by the
other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the remainders
of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline
fun mod(other: UInt): UInt = rem(other)\n\n /**\n * Calculates the remainder of flooring division of this value by
the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the
remainders of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n
public inline fun mod(other: ULong): ULong = this.toULong().mod(other)\n\n /**\n * Returns this value
incremented by one.\n * \n * @sample samples.misc.Builtins.inc\n */\n @kotlin.internal.InlineOnly\n
public inline operator fun inc(): UInt = UInt(data.inc())\n\n /**\n * Returns this value decremented by one.\n
*\n * @sample samples.misc.Builtins.dec\n */\n @kotlin.internal.InlineOnly\n public inline operator fun
dec(): UInt = UInt(data.dec())\n\n /**\n * Creates a range from this value to the specified [other] value. */\n
@kotlin.internal.InlineOnly\n public inline operator fun rangeTo(other: UInt): UIntRange = UIntRange(this,
other)\n\n /**\n * Shifts this value left by the [bitCount] number of bits.\n * \n * Note that only the five
lowest-order bits of the [bitCount] are used as the shift distance.\n * The shift distance actually used is therefore
always in the range `0..31`.\n */\n @kotlin.internal.InlineOnly\n public inline infix fun shl(bitCount: Int): UInt
= UInt(data shl bitCount)\n\n /**\n * Shifts this value right by the [bitCount] number of bits, filling the leftmost
bits with zeros.\n * \n * Note that only the five lowest-order bits of the [bitCount] are used as the shift
distance.\n * The shift distance actually used is therefore always in the range `0..31`.\n */\n
@kotlin.internal.InlineOnly\n public inline infix fun shr(bitCount: Int): UInt = UInt(data ushr bitCount)\n\n /**
Performs a bitwise AND operation between the two values. */\n @kotlin.internal.InlineOnly\n public inline infix
fun and(other: UInt): UInt = UInt(this.data and other.data)\n\n /**\n * Performs a bitwise OR operation between the two
values. */\n @kotlin.internal.InlineOnly\n public inline infix fun or(other: UInt): UInt = UInt(this.data or
other.data)\n\n /**\n * Performs a bitwise XOR operation between the two values. */\n @kotlin.internal.InlineOnly\n
public inline infix fun xor(other: UInt): UInt = UInt(this.data xor other.data)\n\n /**\n * Inverts the bits in this
value. */\n @kotlin.internal.InlineOnly\n public inline fun inv(): UInt = UInt(data.inv())\n\n /**\n * Converts this
[UInt] value to [Byte].\n * \n * If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte`
value represents\n * the same numerical value as this `UInt`.\n * \n * The resulting `Byte` value is
represented by the least significant 8 bits of this `UInt` value.\n * Note that the resulting `Byte` value may be
negative.\n */\n @kotlin.internal.InlineOnly\n public inline fun toByte(): Byte = data.toByte()\n\n /**\n *
Converts this [UInt] value to [Short].\n * \n * If this value is less than or equals to [Short.MAX_VALUE], the
resulting `Short` value represents\n * the same numerical value as this `UInt`.\n * \n * The resulting `Short`
value is represented by the least significant 16 bits of this `UInt` value.\n * Note that the resulting `Short`
value may be negative.\n */\n @kotlin.internal.InlineOnly\n public inline fun toShort(): Short = data.toShort()\n
/**\n * Converts this [UInt] value to [Int].\n * \n * If this value is less than or equals to [Int.MAX_VALUE],
the resulting `Int` value represents\n * the same numerical value as this `UInt`. Otherwise the result is negative.\n
*\n * The resulting `Int` value has the same binary representation as this `UInt` value.\n */\n
@kotlin.internal.InlineOnly\n public inline fun toInt(): Int = data\n\n /**\n * Converts this [UInt] value to
[Long].\n * \n * The resulting `Long` value represents the same numerical value as this `UInt`.\n * \n * The
least significant 32 bits of the resulting `Long` value are the same as the bits of this `UInt` value.\n * whereas the

```

```

most significant 32 bits are filled with zeros.\n
 *^n @kotlin.internal.InlineOnly\n public inline fun toLong():
Long = data.toLong() and 0xFFFF_FFFF\n\n /**\n
 * Converts this [UInt] value to [UByte].\n
 *^n * If this
value is less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents\n
 * the same
numerical value as this `UInt`.\n
 *^n * The resulting `UByte` value is represented by the least significant 8 bits
of this `UInt` value.\n
 *^n @kotlin.internal.InlineOnly\n public inline fun toUByte(): UByte =
data.toUByte()\n /**\n
 * Converts this [UInt] value to [UShort].\n
 *^n * If this value is less than or equals
to [UShort.MAX_VALUE], the resulting `UShort` value represents\n
 * the same numerical value as this `UInt`.\n
 *^n
 * The resulting `UShort` value is represented by the least significant 16 bits of this `UInt` value.\n
 *^n
@kotlin.internal.InlineOnly\n public inline fun toUShort(): UShort = data.toUShort()\n /** Returns this value.
 *^n @kotlin.internal.InlineOnly\n public inline fun toUInt(): UInt = this\n /**\n
 * Converts this [UInt] value
to [ULong].\n
 *^n * The resulting `ULong` value represents the same numerical value as this `UInt`.\n
 *^n
 * The least significant 32 bits of the resulting `ULong` value are the same as the bits of this `UInt` value,\n
 *
whereas the most significant 32 bits are filled with zeros.\n
 *^n @kotlin.internal.InlineOnly\n public inline
fun toULong(): ULong = ULong(data.toLong() and 0xFFFF_FFFF)\n\n /**\n
 * Converts this [UInt] value to
[Float].\n
 *^n * The resulting value is the closest `Float` to this `UInt` value.\n
 * In case when this `UInt`
value is exactly between two `Float`s,\n
 * the one with zero at least significant bit of mantissa is selected.\n
 *^n
@kotlin.internal.InlineOnly\n public inline fun toFloat(): Float = this.toDouble().toFloat()\n /**\n
 * Converts
this [UInt] value to [Double].\n
 *^n * The resulting `Double` value represents the same numerical value as this
`UInt`.\n
 *^n @kotlin.internal.InlineOnly\n public inline fun toDouble(): Double = UIntToDouble(data)\n\n
public override fun toString(): String = toLong().toString()\n\n /**\n
 * Converts this [Byte] value to [UInt].\n
 *^n
 * If this value is positive, the resulting `UInt` value represents the same numerical value as this `Byte`.\n
 *^n
 * The least significant 8 bits of the resulting `UInt` value are the same as the bits of this `Byte` value,\n
 * whereas the
most significant 24 bits are filled with the sign bit of this value.\n
 *^n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n
public inline fun Byte.toUInt(): UInt = UInt(this.toInt())\n\n /**\n
 * Converts this [Short] value to [UInt].\n
 *^n
 * If
this value is positive, the resulting `UInt` value represents the same numerical value as this `Short`.\n
 *^n
 * The least
significant 16 bits of the resulting `UInt` value are the same as the bits of this `Short` value,\n
 * whereas the most
significant 16 bits are filled with the sign bit of this value.\n
 *^n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n
public inline fun Short.toUInt(): UInt = UInt(this.toInt())\n\n /**\n
 * Converts this [Int] value to [UInt].\n
 *^n
 * If this
value is positive, the resulting `UInt` value represents the same numerical value as this `Int`.\n
 *^n
 * The resulting
`UInt` value has the same binary representation as this `Int` value.\n
 *^n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n
public inline fun Int.toUInt(): UInt = UInt(this)\n\n /**\n
 * Converts this [Long] value to [UInt].\n
 *^n
 * If this value
is positive and less than or equals to [UInt.MAX_VALUE], the resulting `UInt` value represents\n
 * the same
numerical value as this `Long`.\n
 *^n
 * The resulting `UInt` value is represented by the least significant 32 bits of
this `Long` value.\n
 *^n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n
public inline fun Long.toUInt(): UInt = UInt(this.toInt())\n\n /**\n
 * Converts this [Float] value to [UInt].\n
 *^n
 *
The fractional part, if any, is rounded down towards zero.\n
 * Returns zero if this `Float` value is negative or `NaN`,
[UInt.MAX_VALUE] if it's bigger than `UInt.MAX_VALUE`.\n
 *^n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n
public inline fun Float.toUInt(): UInt = doubleToUInt(this.toDouble())\n\n /**\n
 * Converts this [Double] value to
[UInt].\n
 *^n
 * The fractional part, if any, is rounded down towards zero.\n
 * Returns zero if this `Double` value is
negative or `NaN`, [UInt.MAX_VALUE] if it's bigger than `UInt.MAX_VALUE`.\n
 *^n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n
public inline fun Double.toUInt(): UInt = doubleToUInt(this)\n\n /**\n
 * Copyright 2010-2022 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n
 * Use of this source code is governed by the Apache 2.0 license that

```

```

can be found in the license/LICENSE.txt file.\n *\n\n// Auto-generated file. DO NOT EDIT!\n\npackage
kotlin\n\nimport kotlin.experimental.*\nimport
kotlin.jvm.*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@JvmInline\npublic value class UShort @kotlin.internal.IntrinsicConstEvaluation @PublishedApi internal
constructor(@PublishedApi internal val data: Short) : Comparable<UShort> {\n\n    companion object {\n\n        /**\n         * A constant holding the minimum value an instance of UShort can have.\n         */\n        public const val
MIN_VALUE: UShort = UShort(0)\n\n        /**\n         * A constant holding the maximum value an instance of
UShort can have.\n         */\n        public const val MAX_VALUE: UShort = UShort(-1)\n\n        /**\n         * The
number of bytes used to represent an instance of UShort in a binary form.\n         */\n        public const val
SIZE_BYTES: Int = 2\n\n        /**\n         * The number of bits used to represent an instance of UShort in a binary
form.\n         */\n        public const val SIZE_BITS: Int = 16\n    }\n\n    /**\n     * Compares this value with the
specified value for order.\n     * Returns zero if this value is equal to the specified other value, a negative number if
it's less than other,\n     * or a positive number if it's greater than other.\n     */\n    @kotlin.internal.InlineOnly\n    public inline operator fun compareTo(other: UByte): Int = this.toInt().compareTo(other.toInt())\n\n    /**\n     *
Compares this value with the specified value for order.\n     * Returns zero if this value is equal to the specified other
value, a negative number if it's less than other,\n     * or a positive number if it's greater than other.\n     */\n    @kotlin.internal.InlineOnly\n    @Suppress("OVERRIDE_BY_INLINE")\n    public override inline operator fun
compareTo(other: UShort): Int = this.toInt().compareTo(other.toInt())\n\n    /**\n     * Compares this value with the
specified value for order.\n     * Returns zero if this value is equal to the specified other value, a negative number if
it's less than other,\n     * or a positive number if it's greater than other.\n     */\n    @kotlin.internal.InlineOnly\n    public inline operator fun compareTo(other: UInt): Int = this.toUInt().compareTo(other)\n\n    /**\n     * Compares
this value with the specified value for order.\n     * Returns zero if this value is equal to the specified other value, a
negative number if it's less than other,\n     * or a positive number if it's greater than other.\n     */\n    @kotlin.internal.InlineOnly\n    public inline operator fun compareTo(other: ULong): Int =
this.toULong().compareTo(other)\n\n    /**\n     * Adds the other value to this value.\n     */\n    @kotlin.internal.InlineOnly\n    public inline operator fun plus(other: UByte): UInt = this.toUInt().plus(other.toUInt())\n\n    /**\n     * Adds the other value
to this value.\n     */\n    @kotlin.internal.InlineOnly\n    public inline operator fun plus(other: UShort): UInt =
this.toUInt().plus(other.toUInt())\n\n    /**\n     * Adds the other value to this value.\n     */\n    @kotlin.internal.InlineOnly\n    public inline operator fun plus(other: UInt): UInt = this.toUInt().plus(other)\n\n    /**\n     * Adds the other value to this
value.\n     */\n    @kotlin.internal.InlineOnly\n    public inline operator fun plus(other: ULong): ULong =
this.toULong().plus(other)\n\n    /**\n     * Subtracts the other value from this value.\n     */\n    @kotlin.internal.InlineOnly\n    public inline operator fun minus(other: UByte): UInt = this.toUInt().minus(other.toUInt())\n\n    /**\n     * Subtracts the
other value from this value.\n     */\n    @kotlin.internal.InlineOnly\n    public inline operator fun minus(other: UShort):
UInt = this.toUInt().minus(other.toUInt())\n\n    /**\n     * Subtracts the other value from this value.\n     */\n    @kotlin.internal.InlineOnly\n    public inline operator fun minus(other: UInt): UInt = this.toUInt().minus(other)\n\n    /**\n     * Subtracts the
other value from this value.\n     */\n    @kotlin.internal.InlineOnly\n    public inline operator fun minus(other: ULong): ULong = this.toULong().minus(other)\n\n    /**\n     * Multiplies this value by the other value.\n     */\n    @kotlin.internal.InlineOnly\n    public inline operator fun times(other: UByte): UInt =
this.toUInt().times(other.toUInt())\n\n    /**\n     * Multiplies this value by the other value.\n     */\n    @kotlin.internal.InlineOnly\n    public inline operator fun times(other: UShort): UInt =
this.toUInt().times(other.toUInt())\n\n    /**\n     * Multiplies this value by the other value.\n     */\n    @kotlin.internal.InlineOnly\n    public inline operator fun times(other: UInt): UInt = this.toUInt().times(other)\n\n    /**\n     * Multiplies this value by the
other value.\n     */\n    @kotlin.internal.InlineOnly\n    public inline operator fun times(other: ULong): ULong = this.toULong().times(other)\n\n    /**\n     * Divides this value by the other value,
truncating the result to an integer that is closer to zero.\n     */\n    @kotlin.internal.InlineOnly\n    public inline operator
fun div(other: UByte): UInt = this.toUInt().div(other.toUInt())\n\n    /**\n     * Divides this value by the other value,
truncating the result to an integer that is closer to zero.\n     */\n    @kotlin.internal.InlineOnly\n    public inline operator
fun div(other: UShort): UInt = this.toUInt().div(other.toUInt())\n\n    /**\n     * Divides this value by the other value,

```



```

truncating the result to an integer that is closer to zero. */
@kotlin.internal.InlineOnly
public inline operator
fun div(other: UInt): UInt = this.toUInt().div(other)
/** Divides this value by the other value, truncating the
result to an integer that is closer to zero. */
@kotlin.internal.InlineOnly
public inline operator fun div(other:
ULong): ULong = this.toULong().div(other)
/**
 * Calculates the remainder of truncating division of this
value by the other value.
 *
 * The result is always less than the divisor. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: UByte): UInt =
this.toUInt().rem(other.toUInt())
/**
 * Calculates the remainder of truncating division of this value by the
other value.
 *
 * The result is always less than the divisor. */
@kotlin.internal.InlineOnly
public
inline operator fun rem(other: UShort): UInt = this.toUInt().rem(other.toUInt())
/**
 * Calculates the
remainder of truncating division of this value by the other value.
 *
 * The result is always less than the
divisor. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: UInt): UInt =
this.toUInt().rem(other)
/**
 * Calculates the remainder of truncating division of this value by the other
value.
 *
 * The result is always less than the divisor. */
@kotlin.internal.InlineOnly
public
inline operator fun rem(other: ULong): ULong = this.toULong().rem(other)
/**
 * Divides this value by
the other value, flooring the result to an integer that is closer to negative infinity.
 *
 * For unsigned types,
the results of flooring division and truncating division are the same. */
@kotlin.internal.InlineOnly
public
inline fun floorDiv(other: UByte): UInt = this.toUInt().floorDiv(other.toUInt())
/**
 * Divides this
value by the other value, flooring the result to an integer that is closer to negative infinity.
 *
 * For unsigned
types, the results of flooring division and truncating division are the same. */
@kotlin.internal.InlineOnly
public
inline fun floorDiv(other: UShort): UInt = this.toUInt().floorDiv(other.toUInt())
/**
 * Divides this
value by the other value, flooring the result to an integer that is closer to negative infinity.
 *
 * For unsigned
types, the results of flooring division and truncating division are the same. */
@kotlin.internal.InlineOnly
public
inline fun floorDiv(other: UInt): UInt = this.toUInt().floorDiv(other)
/**
 * Divides this value by the
other value, flooring the result to an integer that is closer to negative infinity.
 *
 * For unsigned types, the
results of flooring division and truncating division are the same. */
@kotlin.internal.InlineOnly
public
inline fun floorDiv(other: ULong): ULong = this.toULong().floorDiv(other)
/**
 * Calculates the
remainder of flooring division of this value by the other value.
 *
 * The result is always less than the
divisor.
 *
 * For unsigned types, the remainders of flooring division and truncating division are the same. */
@kotlin.internal.InlineOnly
public inline fun mod(other: UByte): UByte =
this.toUInt().mod(other.toUInt()).toUByte()
/**
 * Calculates the remainder of flooring division of this value
by the other value.
 *
 * The result is always less than the divisor.
 *
 * For unsigned types, the
remainders of flooring division and truncating division are the same. */
@kotlin.internal.InlineOnly
public
inline fun mod(other: UShort): UShort = this.toUInt().mod(other.toUInt()).toUShort()
/**
 *
Calculates the remainder of flooring division of this value by the other value.
 *
 * The result is always less
than the divisor.
 *
 * For unsigned types, the remainders of flooring division and truncating division are the
same. */
@kotlin.internal.InlineOnly
public inline fun mod(other: UInt): UInt =
this.toUInt().mod(other)
/**
 * Calculates the remainder of flooring division of this value by the other
value.
 *
 * The result is always less than the divisor.
 *
 * For unsigned types, the remainders of
flooring division and truncating division are the same. */
@kotlin.internal.InlineOnly
public inline fun
mod(other: ULong): ULong = this.toULong().mod(other)
/**
 * Returns this value incremented by one.
 *
 * @sample samples.misc.Builtins.inc
 */
@kotlin.internal.InlineOnly
public inline operator fun
inc(): UShort = UShort(data.inc())
/**
 * Returns this value decremented by one.
 *
 * @sample
samples.misc.Builtins.dec
 */
@kotlin.internal.InlineOnly
public inline operator fun dec(): UShort =
UShort(data.dec())
/**
 * Creates a range from this value to the specified [other] value. */
@kotlin.internal.InlineOnly
public inline operator fun rangeTo(other: UShort): UIntRange =
UIntRange(this.toUInt(), other.toUInt())
/**
 * Performs a bitwise AND operation between the two values. */
@kotlin.internal.InlineOnly
public inline infix fun and(other: UShort): UShort = UShort(this.data and
other.data)
/**
 * Performs a bitwise OR operation between the two values. */
@kotlin.internal.InlineOnly

```

```

public inline infix fun or(other: UShort): UShort = UShort(this.data or other.data)\n /** Performs a bitwise XOR
operation between the two values. */\n @kotlin.internal.InlineOnly\n public inline infix fun xor(other: UShort):
UShort = UShort(this.data xor other.data)\n /** Inverts the bits in this value. */\n @kotlin.internal.InlineOnly\n
public inline fun inv(): UShort = UShort(data.inv())\n\n /**\n * Converts this [UShort] value to [Byte].\n *
* If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value represents\n * the same
numerical value as this `UShort`.\n * The resulting `Byte` value is represented by the least significant 8 bits
of this `UShort` value.\n * Note that the resulting `Byte` value may be negative.\n */\n
@kotlin.internal.InlineOnly\n public inline fun toByte(): Byte = data.toByte()\n /**\n * Converts this [UShort]
value to [Short].\n * If this value is less than or equals to [Short.MAX_VALUE], the resulting `Short` value
represents\n * the same numerical value as this `UShort`. Otherwise the result is negative.\n * The
resulting `Short` value has the same binary representation as this `UShort` value.\n */\n
@kotlin.internal.InlineOnly\n public inline fun toShort(): Short = data\n /**\n * Converts this [UShort] value
to [Int].\n * The resulting `Int` value represents the same numerical value as this `UShort`.\n * The
least significant 16 bits of the resulting `Int` value are the same as the bits of this `UShort` value,\n *
whereas the most significant 16 bits are filled with zeros.\n */\n @kotlin.internal.InlineOnly\n public inline fun toInt(): Int
= data.toInt() and 0xFFFF\n /**\n * Converts this [UShort] value to [Long].\n * The resulting `Long`
value represents the same numerical value as this `UShort`.\n * The least significant 16 bits of the resulting
`Long` value are the same as the bits of this `UShort` value,\n * whereas the most significant 48 bits are filled
with zeros.\n */\n @kotlin.internal.InlineOnly\n public inline fun toLong(): Long = data.toLong() and
0xFFFF\n\n /**\n * Converts this [UShort] value to [UByte].\n * If this value is less than or equals to
[UByte.MAX_VALUE], the resulting `UByte` value represents\n * the same numerical value as this `UShort`.\n
* The resulting `UByte` value is represented by the least significant 8 bits of this `UShort` value.\n */\n
@kotlin.internal.InlineOnly\n public inline fun toUByte(): UByte = data.toUByte()\n /** Returns this value. */\n
@kotlin.internal.InlineOnly\n public inline fun toUShort(): UShort = this\n /**\n * Converts this [UShort]
value to [UInt].\n * The resulting `UInt` value represents the same numerical value as this `UShort`.\n
* The least significant 16 bits of the resulting `UInt` value are the same as the bits of this `UShort` value,\n
* whereas the most significant 16 bits are filled with zeros.\n */\n @kotlin.internal.InlineOnly\n public inline
fun toUInt(): UInt = UInt(data.toInt() and 0xFFFF)\n /**\n * Converts this [UShort] value to [ULong].\n *
The resulting `ULong` value represents the same numerical value as this `UShort`.\n * The least
significant 16 bits of the resulting `ULong` value are the same as the bits of this `UShort` value,\n *
whereas the most significant 48 bits are filled with zeros.\n */\n @kotlin.internal.InlineOnly\n public inline fun toULong():
ULong = ULong(data.toLong() and 0xFFFF)\n\n /**\n * Converts this [UShort] value to [Float].\n *
The resulting `Float` value represents the same numerical value as this `UShort`.\n */\n
@kotlin.internal.InlineOnly\n public inline fun toFloat(): Float = this.toInt().toFloat()\n /**\n * Converts this
[UShort] value to [Double].\n * The resulting `Double` value represents the same numerical value as this
`UShort`.\n */\n @kotlin.internal.InlineOnly\n public inline fun toDouble(): Double =
this.toInt().toDouble()\n\n public override fun toString(): String = toInt().toString()\n\n /**\n * Converts this
[Byte] value to [UShort].\n * If this value is positive, the resulting `UShort` value represents the same numerical
value as this `Byte`.\n * The least significant 8 bits of the resulting `UShort` value are the same as the bits of this
`Byte` value,\n * whereas the most significant 8 bits are filled with the sign bit of this value.\n
*/\n\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n
public inline fun Byte.toUShort(): UShort = UShort(this.toShort())\n /**\n * Converts this [Short] value to
[UShort].\n * If this value is positive, the resulting `UShort` value represents the same numerical value as this
`Short`.\n * The resulting `UShort` value has the same binary representation as this `Short` value.\n
*/\n\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n
public inline fun Short.toUShort(): UShort = UShort(this)\n /**\n * Converts this [Int] value to [UShort].\n *
If this value is positive and less than or equals to [UShort.MAX_VALUE], the resulting `UShort` value represents\n
* the same numerical value as this `Int`.\n * The resulting `UShort` value is represented by the least significant 16

```

bits of this `Int` value.

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Int.toUShort(): UShort = UShort(this.toShort())\n/*\n * Converts this [Long] value to [UShort].\n * If this value is positive and less than or equals to [UShort.MAX_VALUE], the resulting `UShort` value represents the same numerical value as this `Long`.\n * The resulting `UShort` value is represented by the least significant 16 bits of this `Long` value.
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Long.toUShort(): UShort = UShort(this.toShort())\n", /*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
```

```
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("CollectionsKt")\n@file:OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n\npackage kotlin.collections\n\nimport kotlin.contracts.*\nimport kotlin.random.Random\n\ninternal object EmptyIterator : ListIterator<Nothing> {\n    override fun hasNext(): Boolean = false\n    override fun hasPrevious(): Boolean = false\n    override fun nextIndex(): Int = 0\n    override fun previousIndex(): Int = -1\n    override fun next(): Nothing = throw NoSuchElementException()\n    override fun previous(): Nothing = throw NoSuchElementException()\n}\n\ninternal object EmptyList : List<Nothing>, Serializable, RandomAccess {\n    private const val serialVersionUID: Long = -7390468764508069838L\n    override fun equals(other: Any?): Boolean = other is List<*> && other.isEmpty()\n    override fun hashCode(): Int = 1\n    override fun toString(): String = "[]"\n    override val size: Int get() = 0\n    override fun isEmpty(): Boolean = true\n    override fun contains(element: Nothing): Boolean = false\n    override fun containsAll(elements: Collection<Nothing>): Boolean = elements.isEmpty()\n    override fun get(index: Int): Nothing = throw IndexOutOfBoundsException("Empty list doesn't contain element at index $index.")\n    override fun indexOf(element: Nothing): Int = -1\n    override fun lastIndexOf(element: Nothing): Int = -1\n    override fun iterator(): Iterator<Nothing> = EmptyIterator\n    override fun listIterator(): ListIterator<Nothing> = EmptyIterator\n    override fun listIterator(index: Int): ListIterator<Nothing> {\n        if (index != 0) throw IndexOutOfBoundsException("Index: $index")\n        return EmptyIterator\n    }\n    override fun subList(fromIndex: Int, toIndex: Int): List<Nothing> {\n        if (fromIndex == 0 && toIndex == 0) return this\n        throw IndexOutOfBoundsException("fromIndex: $fromIndex, toIndex: $toIndex")\n    }\n    private fun readResolve(): Any = EmptyList\n}\n\ninternal fun <T> Array<out T>.asCollection(): Collection<T> = ArrayAsCollection(this, isVarargs = false)\n\nprivate class ArrayAsCollection<T>(val values: Array<out T>, val isVarargs: Boolean) : Collection<T> {\n    override val size: Int get() = values.size\n    override fun isEmpty(): Boolean = values.isEmpty()\n    override fun contains(element: T): Boolean = values.contains(element)\n    override fun containsAll(elements: Collection<T>): Boolean = elements.all { contains(it) }\n    override fun iterator(): Iterator<T> = values.iterator()\n    // override hidden toArray implementation to prevent copying of values array\n    public fun toArray(): Array<out Any?> = values.copyToArrayOfAny(isVarargs)\n}\n\n/*\n * Returns an empty read-only list. The returned list is serializable (JVM).\n * @sample\nsamples.collections.Collections.Lists.emptyReadOnlyList\n\n*\npublic fun <T> emptyList(): List<T> = EmptyList\n\n/*\n * Returns a new read-only list of given elements. The returned list is serializable (JVM).\n * @sample\nsamples.collections.Collections.Lists.readOnlyList\n\n*\npublic fun <T> listOf(vararg elements: T): List<T> = if (elements.size > 0) elements.asList() else emptyList()\n\n/*\n * Returns an empty read-only list. The returned list is serializable (JVM).\n * @sample\nsamples.collections.Collections.Lists.emptyReadOnlyList\n\n*\n@kotlin.internal.InlineOnly\npublic inline fun <T> listOf(): List<T> = emptyList()\n\n/*\n * Returns an empty new [MutableList].\n * @sample\nsamples.collections.Collections.Lists.emptyMutableList\n\n*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <T> mutableListOf(): MutableList<T> = ArrayList()\n\n/*\n * Returns an empty new [ArrayList].\n * @sample\nsamples.collections.Collections.Lists.emptyArrayList\n\n*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <T> arrayListOf(): ArrayList<T> = ArrayList()\n\n/*\n * Returns a new [MutableList] with the given elements.\n * @sample
```

```

samples.collections.Collections.Lists.mutableList\n *\npublic fun <T> mutableListOf(vararg elements: T):
MutableList<T> =\n    if (elements.size == 0) ArrayList() else ArrayList(ArrayAsCollection(elements, isVarargs =
true))\n\n/**\n * Returns a new [ArrayList] with the given elements.\n * @sample
samples.collections.Collections.Lists.arrayList\n *\npublic fun <T> arrayListOf(vararg elements: T): ArrayList<T>
=\n    if (elements.size == 0) ArrayList() else ArrayList(ArrayAsCollection(elements, isVarargs = true))\n\n/**\n *
Returns a new read-only list either of single given element, if it is not null, or empty list if the element is null. The
returned list is serializable (JVM).\n * @sample samples.collections.Collections.Lists.listOfNotNull\n *\npublic fun
<T : Any> listOfNotNull(element: T?): List<T> = if (element != null) listOf(element) else emptyList()\n\n/**\n *
Returns a new read-only list only of those given elements, that are not null. The returned list is serializable
(JVM).\n * @sample samples.collections.Collections.Lists.listOfNotNull\n *\npublic fun <T : Any>
listOfNotNull(vararg elements: T?): List<T> = elements.filterNotNull()\n\n/**\n * Creates a new read-only list with
the specified [size], where each element is calculated by calling the specified\n * [init] function.\n * \n * The
function [init] is called for each list element sequentially starting from the first one.\n * It should return the value for
a list element given its index.\n * \n * @sample samples.collections.Collections.Lists.readOnlyListFromInitializer\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <T> List(size: Int, init: (index: Int) -> T):
List<T> = MutableList(size, init)\n\n/**\n * Creates a new mutable list with the specified [size], where each element
is calculated by calling the specified\n * [init] function.\n * \n * The function [init] is called for each list element
sequentially starting from the first one.\n * It should return the value for a list element given its index.\n * \n *
@sample samples.collections.Collections.Lists.mutableListFromInitializer\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <T> MutableList(size: Int, init: (index:
Int) -> T): MutableList<T> {\n    val list = ArrayList<T>(size)\n    repeat(size) { index -> list.add(init(index)) }\n
return list}\n\n/**\n * Builds a new read-only [List] by populating a [MutableList] using the given
[builderAction]\n * and returning a read-only list with the same elements.\n * \n * The list passed as a receiver to the
[builderAction] is valid only inside that function.\n * Using it outside of the function produces an unspecified
behavior.\n * \n * The returned list is serializable (JVM).\n * \n * @sample
samples.collections.Builders.Lists.buildListSample\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\n@Su
ppress("DEPRECATION")\npublic inline fun <E> buildList(@BuilderInference builderAction: MutableList<E>().()
-> Unit): List<E> {\n    contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n    return
buildListInternal(builderAction)\n}\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal
expect inline fun <E> buildListInternal(builderAction: MutableList<E>().() -> Unit): List<E>\n\n/**\n * Builds a
new read-only [List] by populating a [MutableList] using the given [builderAction]\n * and returning a read-only list
with the same elements.\n * \n * The list passed as a receiver to the [builderAction] is valid only inside that
function.\n * Using it outside of the function produces an unspecified behavior.\n * \n * The returned list is
serializable (JVM).\n * \n * [capacity] is used to hint the expected number of elements added in the
[builderAction].\n * \n * @throws IllegalArgumentException if the given [capacity] is negative.\n * \n * @sample
samples.collections.Builders.Lists.buildListSampleWithCapacity\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\n@Su
ppress("DEPRECATION")\npublic inline fun <E> buildList(capacity: Int, @BuilderInference builderAction:
MutableList<E>().() -> Unit): List<E> {\n    contract { callsInPlace(builderAction,
InvocationKind.EXACTLY_ONCE) }\n    return buildListInternal(capacity,
builderAction)\n}\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal expect inline
fun <E> buildListInternal(capacity: Int, builderAction: MutableList<E>().() -> Unit): List<E>\n\n/**\n * Returns an
[IntRange] of the valid indices for this collection.\n * @sample
samples.collections.Collections.Collections.indicesOfCollection\n *\npublic val Collection<*>.indices: IntRange\n
get() = 0..size - 1\n\n/**\n * Returns the index of the last item in the list or -1 if the list is empty.\n * \n * @sample
samples.collections.Collections.Lists.lastIndexOfList\n *\npublic val <T> List<T>.lastIndex: Int\n    get() =
this.size - 1\n\n/**\n * Returns `true` if the collection is not empty.\n * @sample

```

```

samples.collections.Collections.Collections.collectionIsNotEmpty\n */\n@kotlin.internal.InlineOnly\npublic inline
fun <T> Collection<T>.isEmpty(): Boolean = !isEmpty()\n\n/**\n * Returns `true` if this nullable collection is
either null or empty.\n * @sample samples.collections.Collections.Collections.collectionOrNull\n
*/\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Collection<T>?.isEmpty():
Boolean {\n    contract {\n        returns(false) implies (this@isEmpty != null)\n    }\n    return this == null ||
this.isEmpty()\n}\n\n/**\n * Returns this Collection if it's not `null` and the empty list otherwise.\n * @sample
samples.collections.Collections.Collections.collectionOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun
<T> Collection<T>?.orEmpty(): Collection<T> = this ?: emptyList()\n\n/**\n * Returns this List if it's not `null` and
the empty list otherwise.\n * @sample samples.collections.Collections.Collections.listOrNull\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun <T> List<T>?.orEmpty(): List<T> = this ?: emptyList()\n\n/**\n * Returns this collection if it's not empty\n * or the result of calling [defaultValue] function if the collection is
empty.\n * @sample samples.collections.Collections.Collections.collectionIfEmpty\n
*/\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <C, R> C.isEmpty(defaultValue: () ->
R): R where C : Collection<*>, C : R =\n    if (isEmpty()) defaultValue() else this\n\n/**\n * Checks if all
elements in the specified collection are contained in this collection.\n * @sample samples.collections.Collections.Collections.collectionContainsAll\n
*/\n@Suppress("EXTENSION_SHADOWED_BY_MEMBER") // false warning, extension takes precedence in
some cases\n@kotlin.internal.InlineOnly\npublic inline fun <@kotlin.internal.OnlyInputTypes T>
Collection<T>.containsAll(elements: Collection<T>): Boolean = this.containsAll(elements)\n\n/**\n * Returns a
new list with the elements of this list randomly shuffled\n * using the specified [random] instance as the source of
randomness.\n * @sample samples.collections.Collections.Collections.collectionShuffled\n
*/\n@SinceKotlin("1.3")\npublic fun <T> Iterable<T>.shuffled(random: Random): List<T> =
toMutableList().apply { shuffle(random) }\n\ninternal fun <T> List<T>.optimizeReadOnlyList() = when (size) {\n
    0 -> emptyList()\n    1 -> listOf(this[0])\n    else -> this\n}\n\n/**\n * Searches this list or its range for the provided
[element] using the binary search algorithm.\n * The list is expected to be sorted into ascending order according to
the Comparable natural ordering of its elements,\n * otherwise the result is undefined.\n * @sample samples.collections.Collections.Collections.Collections.binarySearch\n
*/\n * If the list contains multiple elements equal to the specified [element], there is no guarantee which one will be found.\n * `null`
value is considered to be less than any non-null value.\n * @return the index of the element, if it is contained in
the list within the specified range;\n * otherwise, the inverted insertion point `(-insertion point - 1)`.\n * The
insertion point is defined as the index at which the element should be inserted,\n * so that the list (or the specified
subrange of list) still remains sorted.\n * @sample
samples.collections.Collections.Collections.Collections.binarySearchOnComparable\n * @sample
samples.collections.Collections.Collections.Collections.binarySearchWithBoundaries\n */\npublic fun <T : Comparable<T>>
List<T>?.binarySearch(element: T?, fromIndex: Int = 0, toIndex: Int = size): Int {\n    rangeCheck(size, fromIndex,
toIndex)\n    var low = fromIndex\n    var high = toIndex - 1\n    while (low <= high) {\n        val mid = (low +
high).ushr(1) // safe from overflows\n        val midVal = get(mid)\n        val cmp = compareValues(midVal,
element)\n        if (cmp < 0)\n            low = mid + 1\n        else if (cmp > 0)\n            high = mid - 1\n        else\n
            return mid // key found\n    }\n    return -(low + 1) // key not found\n}\n\n/**\n * Searches this list or its range
for the provided [element] using the binary search algorithm.\n * The list is expected to be sorted into ascending
order according to the specified [comparator],\n * otherwise the result is undefined.\n * @sample samples.collections.Collections.Collections.Collections.binarySearchWithComparator\n
*/\n * If the list contains multiple elements equal to the specified [element], there is no guarantee which one will be found.\n * `null`
value is considered to be less than any non-null value.\n * @return the index of the element, if it is contained in
the list within the specified range;\n * otherwise, the inverted insertion point `(-insertion point - 1)`.\n * The
insertion point is defined as the index at which the element should be inserted,\n * so that the list (or the specified
subrange of list) still remains sorted according to the specified [comparator].\n * @sample
samples.collections.Collections.Collections.Collections.binarySearchWithComparator\n */\npublic fun <T>
List<T>.binarySearch(element: T, comparator: Comparator<in T>, fromIndex: Int = 0, toIndex: Int = size): Int {\n
    rangeCheck(size, fromIndex, toIndex)\n    var low = fromIndex\n    var high = toIndex - 1\n    while (low <=

```

```

high) {\n    val mid = (low + high).ushr(1) // safe from overflows\n    val midVal = get(mid)\n    val cmp =
comparator.compare(midVal, element)\n    if (cmp < 0)\n        low = mid + 1\n    else if (cmp > 0)\n        high = mid - 1\n    else\n        return mid // key found\n    }\n    return -(low + 1) // key not found\n}\n\n/**\n * Searches this list or its range for an element having the key returned by the specified [selector] function\n * equal to the provided [key] value using the binary search algorithm.\n * The list is expected to be sorted into ascending order according to the Comparable natural ordering of keys of its elements.\n * otherwise the result is undefined.\n * *\n * If the list contains multiple elements with the specified [key], there is no guarantee which one will be found.\n * *\n * `null` value is considered to be less than any non-null value.\n * *\n * @return the index of the element with the specified [key], if it is contained in the list within the specified range;\n * otherwise, the inverted insertion point `(-insertion point - 1)`.\n * The insertion point is defined as the index at which the element should be inserted,\n * so that the list (or the specified subrange of list) still remains sorted.\n * *\n * @sample
samples.collections.Collections.Lists.binarySearchByKey\n */\npublic inline fun <T, K : Comparable<K>>
List<T>.binarySearchBy(\n    key: K?,\n    fromIndex: Int = 0,\n    toIndex: Int = size,\n    crossinline selector: (T) ->
K?)\n): Int =\n    binarySearch(fromIndex, toIndex) { compareValues(selector(it), key) }\n\n// do not introduce this
overload --- too rare\n//\npublic fun <T, K> List<T>.binarySearchBy(key: K, comparator: Comparator<K>,
fromIndex: Int = 0, toIndex: Int = size(), selector: (T) -> K): Int =\n    binarySearch(fromIndex, toIndex) {
comparator.compare(selector(it), key) }\n\n\n/**\n * Searches this list or its range for an element for which the
given [comparison] function returns zero using the binary search algorithm.\n * *\n * The list is expected to be sorted
so that the signs of the [comparison] function's return values ascend on the list elements,\n * i.e. negative values
come before zero and zeroes come before positive values.\n * *\n * Otherwise, the result is undefined.\n * *\n * If the list
contains multiple elements for which [comparison] returns zero, there is no guarantee which one will be found.\n * *\n *
@param comparison function that returns zero when called on the list element being searched.\n * *\n * On the
elements coming before the target element, the function must return negative values;\n * on the elements coming
after the target element, the function must return positive values.\n * *\n * @return the index of the found element, if
it is contained in the list within the specified range;\n * otherwise, the inverted insertion point `(-insertion point -
1)`.\n * The insertion point is defined as the index at which the element should be inserted,\n * so that the list (or the
specified subrange of list) still remains sorted.\n * *\n * @sample
samples.collections.Collections.Lists.binarySearchWithComparisonFunction\n */\npublic fun <T>
List<T>.binarySearch(fromIndex: Int = 0, toIndex: Int = size, comparison: (T) -> Int): Int {\n    rangeCheck(size,
fromIndex, toIndex)\n    var low = fromIndex\n    var high = toIndex - 1\n    while (low <= high) {\n        val mid
= (low + high).ushr(1) // safe from overflows\n        val midVal = get(mid)\n        val cmp = comparison(midVal)\n        if (cmp < 0)\n            low = mid + 1\n        else if (cmp > 0)\n            high = mid - 1\n        else\n            return
mid // key found\n    }\n    return -(low + 1) // key not found\n}\n\n\n/**\n * Checks that `from` and `to` are in\n * the
range of [0..size] and throws an appropriate exception, if they aren't.\n * *\n * @private fun rangeCheck(size: Int,
fromIndex: Int, toIndex: Int) {\n    when {\n        fromIndex > toIndex -> throw
IllegalArgumentException("fromIndex ($fromIndex) is greater than toIndex ($toIndex).")\n        fromIndex < 0 ->
throw IndexOutOfBoundsException("fromIndex ($fromIndex) is less than zero.")\n        toIndex > size -> throw
IndexOutOfBoundsException("toIndex ($toIndex) is greater than size ($size).")\n    }\n}\n\n\n@PublishedApi\n@SinceKotlin("1.3")\ninternal expect fun checkIndexOverflow(index: Int):
Int\n\n@PublishedApi\n@SinceKotlin("1.3")\ninternal expect fun checkCountOverflow(count: Int):
Int\n\n\n@PublishedApi\n@SinceKotlin("1.3")\ninternal fun throwIndexOverflow() { throw
ArithmeticException("Index overflow has happened.") }\n\n@PublishedApi\n@SinceKotlin("1.3")\ninternal fun
throwCountOverflow() { throw ArithmeticException("Count overflow has happened.") }\n\n\n/**\n * Copyright
2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *
*\n * @file:kotlin.jvm.JvmMultifileClass\n * @file:kotlin.jvm.JvmName("MapsKt")\n * @file:OptIn(kotlin.experimental.
ExperimentalTypeInference::class)\n * @package kotlin.collections\n * @import kotlin.contracts.*\n * @private object
EmptyMap : Map<Any?, Nothing>, Serializable {\n    private const val serialVersionUID: Long =

```

```

8246714829545688274\n\n override fun equals(other: Any?): Boolean = other is Map<*, *> &&
other.isEmpty()\n\n override fun hashCode(): Int = 0\n\n override fun toString(): String = "{}"\n\n override val
size: Int get() = 0\n\n override fun isEmpty(): Boolean = true\n\n\n override fun containsKey(key: Any?): Boolean =
false\n\n\n override fun containsValue(value: Nothing): Boolean = false\n\n\n override fun get(key: Any?): Nothing? =
null\n\n\n override val entries: Set<Map.Entry<Any?, Nothing>> get() = EmptySet\n\n\n override val keys: Set<Any?>
get() = EmptySet\n\n\n override val values: Collection<Nothing> get() = EmptyList\n\n\n private fun readResolve():
Any = EmptyMap\n\n}\n\n/**\n * Returns an empty read-only map of specified type.\n * \n * The returned map is
serializable (JVM).\n * @sample samples.collections.Maps.Instantiation.emptyReadOnlyMap\n */\n\npublic fun <K,
V> emptyMap(): Map<K, V> = @Suppress("UNCHECKED_CAST") (EmptyMap as Map<K, V>)\n\n/**\n *
Returns a new read-only map with the specified contents, given as a list of pairs\n * where the first value is the key
and the second is the value.\n * \n * If multiple pairs have the same key, the resulting map will contain the value
from the last of those pairs.\n * \n * Entries of the map are iterated in the order they were specified.\n * \n * The
returned map is serializable (JVM).\n * \n * @sample samples.collections.Maps.Instantiation.mapFromPairs\n */\n\npublic fun <K, V> mapOf(vararg pairs: Pair<K, V>): Map<K, V> =\n\n if (pairs.size > 0)
pairs.toMap(LinkedHashMap(mapCapacity(pairs.size))) else emptyMap()\n\n/**\n * Returns an empty read-only
map.\n * \n * The returned map is serializable (JVM).\n * @sample
samples.collections.Maps.Instantiation.emptyReadOnlyMap\n */\n\n@kotlin.internal.InlineOnly\n\npublic inline fun
<K, V> mapOf(): Map<K, V> = emptyMap()\n\n/**\n * Returns an empty new [MutableMap].\n * \n * The returned
map preserves the entry iteration order.\n * @sample samples.collections.Maps.Instantiation.emptyMutableMap\n */\n\n@SinceKotlin("1.1")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <K, V> mutableMapOf():
MutableMap<K, V> = LinkedHashMap()\n\n/**\n * Returns a new [MutableMap] with the specified contents, given
as a list of pairs\n * where the first component is the key and the second is the value.\n * \n * If multiple pairs have
the same key, the resulting map will contain the value from the last of those pairs.\n * \n * Entries of the map are
iterated in the order they were specified.\n * \n * @sample
samples.collections.Maps.Instantiation.mutableMapFromPairs\n * @sample
samples.collections.Maps.Instantiation.emptyMutableMap\n */\n\npublic fun <K, V> mutableMapOf(vararg pairs:
Pair<K, V>): MutableMap<K, V> =\n\n LinkedHashMap<K, V>(mapCapacity(pairs.size)).apply { putAll(pairs)
}\n\n/**\n * Returns an empty new [HashMap].\n * \n * @sample
samples.collections.Maps.Instantiation.emptyHashMap\n */\n\n@SinceKotlin("1.1")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <K, V> hashMapOf(): HashMap<K, V>
= HashMap<K, V>()\n\n/**\n * Returns a new [HashMap] with the specified contents, given as a list of pairs\n *
where the first component is the key and the second is the value.\n * \n * @sample
samples.collections.Maps.Instantiation.hashMapFromPairs\n */\n\npublic fun <K, V> hashMapOf(vararg pairs:
Pair<K, V>): HashMap<K, V> = HashMap<K, V>(mapCapacity(pairs.size)).apply { putAll(pairs) }\n\n/**\n *
Returns an empty new [LinkedHashMap].\n * \n * @SinceKotlin("1.1")\n */\n\n@kotlin.internal.InlineOnly\n\npublic inline
fun <K, V> linkedMapOf(): LinkedHashMap<K, V> = LinkedHashMap<K, V>()\n\n/**\n * Returns a new
[LinkedHashMap] with the specified contents, given as a list of pairs\n * where the first component is the key and
the second is the value.\n * \n * If multiple pairs have the same key, the resulting map will contain the value from the
last of those pairs.\n * \n * Entries of the map are iterated in the order they were specified.\n * \n * @sample
samples.collections.Maps.Instantiation.linkedMapFromPairs\n */\n\npublic fun <K, V> linkedMapOf(vararg pairs:
Pair<K, V>): LinkedHashMap<K, V> = pairs.toMap(LinkedHashMap(mapCapacity(pairs.size)))\n\n/**\n * Builds
a new read-only [Map] by populating a [MutableMap] using the given [builderAction]\n * and returning a read-only
map with the same key-value pairs.\n * \n * The map passed as a receiver to the [builderAction] is valid only inside
that function.\n * Using it outside of the function produces an unspecified behavior.\n * \n * Entries of the map are
iterated in the order they were added by the [builderAction].\n * \n * The returned map is serializable (JVM).\n * \n *
@sample samples.collections.Builders.Maps.buildMapSample\n */\n\n@SinceKotlin("1.6")\n\n@WasExperimental(ExperimentalStdlibApi::class)\n\n@kotlin.internal.InlineOnly\n\n@Suppress("DEPRECATION")\n\npublic inline fun <K, V> buildMap(@BuilderInference builderAction:

```

```

MutableMap<K, V>().->Unit): Map<K, V> { \n  contract { callsInPlace(builderAction,
InvocationKind.EXACTLY_ONCE) } \n  return
buildMapInternal(builderAction)\n}\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal expect inline fun <K, V> buildMapInternal(builderAction: MutableMap<K, V>().->Unit): Map<K,
V>\n\n/** \n * Builds a new read-only [Map] by populating a [MutableMap] using the given [builderAction] \n * and
returning a read-only map with the same key-value pairs. \n * \n * The map passed as a receiver to the
[builderAction] is valid only inside that function. \n * Using it outside of the function produces an unspecified
behavior. \n * \n * [capacity] is used to hint the expected number of pairs added in the [builderAction]. \n * \n * Entries
of the map are iterated in the order they were added by the [builderAction]. \n * \n * The returned map is serializable
(JVM). \n * \n * @throws IllegalArgumentException if the given [capacity] is negative. \n * \n * @sample
samples.collections.Builders.Builders.buildMapSample\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\n@Su
ppress("DEPRECATION")\npublic inline fun <K, V> buildMap(capacity: Int, @BuilderInference builderAction:
MutableMap<K, V>().->Unit): Map<K, V> { \n  contract { callsInPlace(builderAction,
InvocationKind.EXACTLY_ONCE) } \n  return buildMapInternal(capacity,
builderAction)\n}\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal expect inline
fun <K, V> buildMapInternal(capacity: Int, builderAction: MutableMap<K, V>().->Unit): Map<K, V>\n\n/** \n *
Calculate the initial capacity of a map. \n *\n@PublishedApi\ninternal expect fun mapCapacity(expectedSize: Int):
Int\n\n/** \n * Returns `true` if this map is not empty. \n * \n * @sample
samples.collections.Maps.Usage.mapIsNotEmpty\n *\n@kotlin.internal.InlineOnly\npublic inline fun <K, V>
Map<out K, V>.isEmpty(): Boolean = !isEmpty()\n\n/** \n * Returns `true` if this nullable map is either null or
empty. \n * \n * @sample samples.collections.Maps.Usage.mapIsNullOrEmpty\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K,
V>?.isNullOrEmpty(): Boolean { \n  contract { \n    returns(false) implies (this@isNullOrEmpty != null)\n  } \n\n
return this == null || isEmpty()\n}\n\n/** \n * Returns the [Map] if its not `null`, or the empty [Map] otherwise. \n
*\n * @sample samples.collections.Maps.Usage.mapOrElse\n *\n@kotlin.internal.InlineOnly\npublic inline fun
<K, V> Map<K, V>?.orElse(): Map<K, V> = this ?: emptyMap()\n\n/** \n * Returns this map if it's not empty \n *
or the result of calling [defaultValue] function if the map is empty. \n * \n * @sample
samples.collections.Maps.Usage.mapIfEmpty\n *\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic
inline fun <M, R> M.ifEmpty(defaultValue: () -> R): R where M : Map<*, *>, M : R = \n  if (isEmpty())
defaultValue() else this\n\n/** \n * Checks if the map contains the given key. \n * \n * This method allows to use the
`x in map` syntax for checking whether an object is contained in the map. \n * \n * @sample
samples.collections.Maps.Usage.containsKey\n *\n@kotlin.internal.InlineOnly\npublic inline operator fun
<@kotlin.internal.OnlyInputTypes K, V> Map<out K, V>.contains(key: K): Boolean = containsKey(key)\n\n/** \n *
Returns the value corresponding to the given [key], or `null` if such a key is not present in the map. \n
*\n *\n@kotlin.internal.InlineOnly\npublic inline operator fun <@kotlin.internal.OnlyInputTypes K, V> Map<out K,
V>.get(key: K): V? = \n  @Suppress("UNCHECKED_CAST") (this as Map<K, V>).get(key)\n\n/** \n * Allows
to use the index operator for storing values in a mutable map. \n *\n@kotlin.internal.InlineOnly\npublic inline
operator fun <K, V> MutableMap<K, V>.set(key: K, value: V): Unit { \n  put(key, value)\n}\n\n/** \n * Returns
`true` if the map contains the specified [key]. \n * \n * Allows to overcome type-safety restriction of `containsKey`
that requires to pass a key of type `K`. \n *\n@kotlin.internal.InlineOnly\npublic inline fun
<@kotlin.internal.OnlyInputTypes K> Map<out K, *>.containsKey(key: K): Boolean = \n
@Suppress("UNCHECKED_CAST") (this as Map<K, *>).containsKey(key)\n\n/** \n * Returns `true` if the map
maps one or more keys to the specified [value]. \n * \n * Allows to overcome type-safety restriction of
`containsValue` that requires to pass a value of type `V`. \n * \n * @sample
samples.collections.Maps.Usage.containsValue\n *\n@Suppress("EXTENSION_SHADOWED_BY_MEMBER")
// false warning, extension takes precedence in some cases\n@kotlin.internal.InlineOnly\npublic inline fun <K,
@kotlin.internal.OnlyInputTypes V> Map<K, V>.containsValue(value: V): Boolean =

```


`this.containsValue(value)`
 Removes the specified key and its corresponding value from this map.
 Returns the previous value associated with the key, or `null` if the key was not present in the map.
 Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.
`@kotlin.internal.InlineOnly`
`public inline fun <@kotlin.internal.OnlyInputTypes K, V> MutableMap<out K, V>.remove(key: K): V? = @Suppress("UNCHECKED_CAST") (this as MutableMap<K, V>).remove(key)`
 Returns the key component of the map entry.
 This method allows to use destructuring declarations when working with maps, for example:
`for ((key, value) in map) { // do something with the key and the value }`
`@kotlin.internal.InlineOnly`
`public inline operator fun <K, V> Map.Entry<K, V>.component1(): K = key`
 Returns the value component of the map entry.
 This method allows to use destructuring declarations when working with maps, for example:
`for ((key, value) in map) { // do something with the key and the value }`
`@kotlin.internal.InlineOnly`
`public inline operator fun <K, V> Map.Entry<K, V>.component2(): V = value`
 Converts entry to `[Pair]` with key being first component and value being second.
`@kotlin.internal.InlineOnly`
`public inline fun <K, V> Map.Entry<K, V>.toPair(): Pair<K, V> = Pair(key, value)`
 Returns the value for the given key, or the result of the `[defaultValue]` function if there was no entry for the given key.
`@sample` `samples.collections.Maps.Usage.getOrNull`
`@kotlin.internal.InlineOnly`
`public inline fun <K, V> Map<K, V>.getOrNull(key: K, defaultValue: () -> V): V = get(key) ?: defaultValue()`
`internal inline fun <K, V> Map<K, V>.getOrNullNullable(key: K, defaultValue: () -> V): V { val value = get(key) if (value == null && !containsKey(key)) { return defaultValue() } else { @Suppress("UNCHECKED_CAST") return value as V } }`
 Returns the value for the given `[key]` or throws an exception if there is no such key in the map.
 If the map was created by `[withDefault]`, resorts to its `defaultValue` provider function instead of throwing an exception.
 Throws `NoSuchElementException` when the map doesn't contain a value for the specified key and no implicit default value was provided for that map.
`@SinceKotlin("1.1")`
`public fun <K, V> Map<K, V>.getValue(key: K): V = getOrImplicitDefault(key)`
 Returns the value for the given key. If the key is not found in the map, calls the `[defaultValue]` function, puts its result into the map under the given key and returns it.
 Note that the operation is not guaranteed to be atomic if the map is being modified concurrently.
`@sample` `samples.collections.Maps.Usage.getOrPut`
`public inline fun <K, V> MutableMap<K, V>.getOrPut(key: K, defaultValue: () -> V): V { val value = get(key) return if (value == null) { val answer = defaultValue() put(key, answer) answer } else { value } }`
 Returns an `[Iterator]` over the entries in the `[Map]`.
`@sample` `samples.collections.Maps.Usage.forOverEntries`
`@kotlin.internal.InlineOnly`
`public inline operator fun <K, V> Map<out K, V>.iterator(): Iterator<Map.Entry<K, V>> = entries.iterator()`
 Returns a `[MutableIterator]` over the mutable entries in the `[MutableMap]`.
`@kotlin.jvm.JvmName("mutableIterator")`
`@kotlin.internal.InlineOnly`
`public inline operator fun <K, V> MutableMap<K, V>.iterator(): MutableIterator<MutableMap.MutableEntry<K, V>> = entries.iterator()`
 Populates the given `[destination]` map with entries having the keys of this map and the values obtained by applying the `[transform]` function to each entry in this `[Map]`.
`public inline fun <K, V, R, M : MutableMap<in K, in R>> Map<out K, V>.mapValuesTo(destination: M, transform: (Map.Entry<K, V>) -> R): M { return entries.associateByTo(destination, { it.key }, transform) }`
 Populates the given `[destination]` map with entries having the keys obtained by applying the `[transform]` function to each entry in this `[Map]` and the values of this map.
 In case if any two entries are mapped to the equal keys, the value of the latter one will overwrite the value associated with the former one.
`public inline fun <K, V, R, M : MutableMap<in R, in V>> Map<out K, V>.mapKeysTo(destination: M, transform: (Map.Entry<K, V>) -> R): M { return entries.associateByTo(destination, transform, { it.value }) }`
 Puts all the given `[pairs]` into this `[MutableMap]` with the first component in the pair being the key and the second the value.
`public fun <K, V> MutableMap<in K, in V>.putAll(pairs: Array<out Pair<K, V>>): Unit { for ((key, value) in pairs) { put(key, value) } }`
 Puts all the elements of the given collection into this `[MutableMap]` with the first component in the pair being the key and the second the value.
`public fun <K, V> MutableMap<in K, in`

```

V>.putAll(pairs: Iterable<Pair<K, V>>): Unit {
    for ((key, value) in pairs) {
        put(key, value)
    }
}

* Puts all the elements of the given sequence into this [MutableMap] with the first component in the pair being the key and the second the value.

public fun <K, V> MutableMap<in K, in V>.putAll(pairs: Sequence<Pair<K, V>>): Unit {
    for ((key, value) in pairs) {
        put(key, value)
    }
}

* Returns a new map with entries having the keys of this map and the values obtained by applying the [transform] function to each entry in this [Map].

* The returned map preserves the entry iteration order of the original map.

@sample samples.collections.Maps.Transformations.mapValues

public inline fun <K, V, R> Map<out K, V>.mapValues(transform: (Map.Entry<K, V>) -> R): Map<K, R> {
    return mapValuesTo(LinkedHashMap<K, R>(mapCapacity(size)), transform) // .optimizeReadOnlyMap()
}

* Returns a new Map with entries having the keys obtained by applying the [transform] function to each entry in this [Map] and the values of this map.

* In case if any two entries are mapped to the equal keys, the value of the latter one will overwrite the value associated with the former one.

* The returned map preserves the entry iteration order of the original map.

@sample samples.collections.Maps.Transformations.mapKeys

public inline fun <K, V, R> Map<out K, V>.mapKeys(transform: (Map.Entry<K, V>) -> R): Map<R, V> {
    return mapKeysTo(LinkedHashMap<R, V>(mapCapacity(size)), transform) // .optimizeReadOnlyMap()
}

* Returns a map containing all key-value pairs with keys matching the given [predicate].

* The returned map preserves the entry iteration order of the original map.

@sample samples.collections.Maps.Filtering.filterKeys

public inline fun <K, V> Map<out K, V>.filterKeys(predicate: (K) -> Boolean): Map<K, V> {
    val result = LinkedHashMap<K, V>()
    for (entry in this) {
        if (predicate(entry.key)) {
            result.put(entry.key, entry.value)
        }
    }
    return result
}

* Returns a map containing all key-value pairs with values matching the given [predicate].

* The returned map preserves the entry iteration order of the original map.

@sample samples.collections.Maps.Filtering.filterValues

public inline fun <K, V> Map<out K, V>.filterValues(predicate: (V) -> Boolean): Map<K, V> {
    val result = LinkedHashMap<K, V>()
    for (entry in this) {
        if (predicate(entry.value)) {
            result.put(entry.key, entry.value)
        }
    }
    return result
}

* Appends all entries matching the given [predicate] into the mutable map given as [destination] parameter.

* @return the destination map.

@sample samples.collections.Maps.Filtering.filterTo

public inline fun <K, V, M : MutableMap<in K, in V>> Map<out K, V>.filterTo(destination: M, predicate: (Map.Entry<K, V>) -> Boolean): M {
    for (element in this) {
        if (predicate(element)) {
            destination.put(element.key, element.value)
        }
    }
    return destination
}

* Returns a new map containing all key-value pairs matching the given [predicate].

* The returned map preserves the entry iteration order of the original map.

@sample samples.collections.Maps.Filtering.filter

public inline fun <K, V> Map<out K, V>.filter(predicate: (Map.Entry<K, V>) -> Boolean): Map<K, V> {
    return filterTo(LinkedHashMap<K, V>(), predicate)
}

* Appends all entries not matching the given [predicate] into the given [destination].

* @return the destination map.

@sample samples.collections.Maps.Filtering.filterNotTo

public inline fun <K, V, M : MutableMap<in K, in V>> Map<out K, V>.filterNotTo(destination: M, predicate: (Map.Entry<K, V>) -> Boolean): M {
    for (element in this) {
        if (!predicate(element)) {
            destination.put(element.key, element.value)
        }
    }
    return destination
}

* Returns a new map containing all key-value pairs not matching the given [predicate].

* The returned map preserves the entry iteration order of the original map.

@sample samples.collections.Maps.Filtering.filterNot

public inline fun <K, V> Map<out K, V>.filterNot(predicate: (Map.Entry<K, V>) -> Boolean): Map<K, V> {
    return filterNotTo(LinkedHashMap<K, V>(), predicate)
}

* Returns a new map containing all key-value pairs from the given collection of pairs.

* The returned map preserves the entry iteration order of the original collection.

* If any of two pairs would have the same key the last one gets added to the map.

public fun <K, V> Iterable<Pair<K, V>>.toMap(): Map<K, V> {
    if (this is Collection) {
        return when (size) {
            0 -> emptyMap()
            1 -> mapOf(if (this is List) this[0] else iterator().next())
            else -> toMap(LinkedHashMap<K, V>(mapCapacity(size)))
        }
    }
    return toMap(LinkedHashMap<K, V>()).optimizeReadOnlyMap()
}

* Populates and returns the [destination] mutable map with key-value pairs from the given collection of pairs.

public fun <K, V, M :

```

```

MutableMap<in K, in V>> Iterable<Pair<K, V>>.toMap(destination: M): M =\n  destination.apply {
putAll(this@toMap) }\n\n/**\n * Returns a new map containing all key-value pairs from the given array of pairs.\n
*\n * The returned map preserves the entry iteration order of the original array.\n * If any of two pairs would have
the same key the last one gets added to the map.\n */\npublic fun <K, V> Array<out Pair<K, V>>.toMap(): Map<K,
V> = when (size) {\n  0 -> emptyMap()\n  1 -> mapOf(this[0])\n  else -> toMap(LinkedHashMap<K,
V>(mapCapacity(size)))\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs
from the given array of pairs.\n */\npublic fun <K, V, M : MutableMap<in K, in V>> Array<out Pair<K,
V>>.toMap(destination: M): M =\n  destination.apply { putAll(this@toMap) }\n\n/**\n * Returns a new map
containing all key-value pairs from the given sequence of pairs.\n *\n * The returned map preserves the entry
iteration order of the original sequence.\n * If any of two pairs would have the same key the last one gets added to
the map.\n */\npublic fun <K, V> Sequence<Pair<K, V>>.toMap(): Map<K, V> = toMap(LinkedHashMap<K,
V>()).optimizeReadOnlyMap()\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs
from the given sequence of pairs.\n */\npublic fun <K, V, M : MutableMap<in K, in V>> Sequence<Pair<K,
V>>.toMap(destination: M): M =\n  destination.apply { putAll(this@toMap) }\n\n/**\n * Returns a new read-only
map containing all key-value pairs from the original map.\n *\n * The returned map preserves the entry iteration
order of the original map.\n */\n@SinceKotlin("1.1")\npublic fun <K, V> Map<out K, V>.toMap(): Map<K, V> =
when (size) {\n  0 -> emptyMap()\n  1 -> toSingletonMap()\n  else -> toMutableMap()\n}\n\n/**\n * Returns a
new mutable map containing all key-value pairs from the original map.\n *\n * The returned map preserves the entry
iteration order of the original map.\n */\n@SinceKotlin("1.1")\npublic fun <K, V> Map<out K,
V>.toMutableMap(): MutableMap<K, V> = LinkedHashMap(this)\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs from the given map.\n */\n@SinceKotlin("1.1")\npublic fun <K, V,
M : MutableMap<in K, in V>> Map<out K, V>.toMap(destination: M): M =\n  destination.apply {
putAll(this@toMap) }\n\n/**\n * Creates a new read-only map by replacing or adding an entry to this map from a
given key-value [pair].\n *\n * The returned map preserves the entry iteration order of the original map.\n * The
[pair] is iterated in the end if it has a unique key.\n */\npublic operator fun <K, V> Map<out K, V>.plus(pair:
Pair<K, V>): Map<K, V> =\n  if (this.isEmpty()) mapOf(pair) else LinkedHashMap(this).apply { put(pair.first,
pair.second) }\n\n/**\n * Creates a new read-only map by replacing or adding entries to this map from a given
collection of key-value [pairs].\n *\n * The returned map preserves the entry iteration order of the original map.\n
* Those [pairs] with unique keys are iterated in the end in the order of [pairs] collection.\n */\npublic operator fun <K,
V> Map<out K, V>.plus(pairs: Iterable<Pair<K, V>>): Map<K, V> =\n  if (this.isEmpty()) pairs.toMap() else
LinkedHashMap(this).apply { putAll(pairs) }\n\n/**\n * Creates a new read-only map by replacing or adding entries
to this map from a given array of key-value [pairs].\n *\n * The returned map preserves the entry iteration order of
the original map.\n * Those [pairs] with unique keys are iterated in the end in the order of [pairs] array.\n */\npublic
operator fun <K, V> Map<out K, V>.plus(pairs: Array<out Pair<K, V>>): Map<K, V> =\n  if (this.isEmpty())
pairs.toMap() else LinkedHashMap(this).apply { putAll(pairs) }\n\n/**\n * Creates a new read-only map by
replacing or adding entries to this map from a given sequence of key-value [pairs].\n *\n * The returned map
preserves the entry iteration order of the original map.\n * Those [pairs] with unique keys are iterated in the end in
the order of [pairs] sequence.\n */\npublic operator fun <K, V> Map<out K, V>.plus(pairs: Sequence<Pair<K,
V>>): Map<K, V> =\n  LinkedHashMap(this).apply { putAll(pairs) }.optimizeReadOnlyMap()\n\n/**\n * Creates
a new read-only map by replacing or adding entries to this map from another [map].\n *\n * The returned map
preserves the entry iteration order of the original map.\n * Those entries of another [map] that are missing in this
map are iterated in the end in the order of that [map].\n */\npublic operator fun <K, V> Map<out K, V>.plus(map:
Map<out K, V>): Map<K, V> =\n  LinkedHashMap(this).apply { putAll(map) }\n\n/**\n * Appends or replaces
the given [pair] in this mutable map.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V>
MutableMap<in K, in V>.plusAssign(pair: Pair<K, V>) {\n  put(pair.first, pair.second)\n}\n\n/**\n * Appends or
replaces all pairs from the given collection of [pairs] in this mutable map.\n */\n@kotlin.internal.InlineOnly\npublic
inline operator fun <K, V> MutableMap<in K, in V>.plusAssign(pairs: Iterable<Pair<K, V>>) {\n
  putAll(pairs)\n}\n\n/**\n * Appends or replaces all pairs from the given array of [pairs] in this mutable map.\n

```

```

*  

@kotlin.internal.InlineOnly  

public inline operator fun <K, V> MutableMap<in K, in V>.plusAssign(pairs:  

Array<out Pair<K, V>>) {  

    putAll(pairs)  

}
* Appends or replaces all pairs from the given sequence of  

[pairs] in this mutable map.
*  

@kotlin.internal.InlineOnly  

public inline operator fun <K, V> MutableMap<in  

K, in V>.plusAssign(pairs: Sequence<Pair<K, V>>) {  

    putAll(pairs)  

}
* Appends or replaces all  

entries from the given [map] in this mutable map.
*  

@kotlin.internal.InlineOnly  

public inline operator fun <K,  

V> MutableMap<in K, in V>.plusAssign(map: Map<K, V>) {  

    putAll(map)  

}
* Returns a map  

containing all entries of the original map except the entry with the given [key].
* The returned map preserves  

the entry iteration order of the original map.
*  

@SinceKotlin("1.1")  

public operator fun <K, V> Map<out K,  

V>.minus(key: K): Map<K, V> =  

    this.toMutableMap().apply { minusAssign(key)  

}.optimizeReadOnlyMap()
* Returns a map containing all entries of the original map except those entries  

* the keys of which are contained in the given [keys] collection.
* The returned map preserves the entry  

iteration order of the original map.
*  

@SinceKotlin("1.1")  

public operator fun <K, V> Map<out K,  

V>.minus(keys: Iterable<K>): Map<K, V> =  

    this.toMutableMap().apply { minusAssign(keys)  

}.optimizeReadOnlyMap()
* Returns a map containing all entries of the original map except those entries  

* the keys of which are contained in the given [keys] array.
* The returned map preserves the entry iteration  

order of the original map.
*  

@SinceKotlin("1.1")  

public operator fun <K, V> Map<out K, V>.minus(keys:  

Array<out K>): Map<K, V> =  

    this.toMutableMap().apply { minusAssign(keys)  

}.optimizeReadOnlyMap()
* Returns a map containing all entries of the original map except those entries  

* the keys of which are contained in the given [keys] sequence.
* The returned map preserves the entry  

iteration order of the original map.
*  

@SinceKotlin("1.1")  

public operator fun <K, V> Map<out K,  

V>.minus(keys: Sequence<K>): Map<K, V> =  

    this.toMutableMap().apply { minusAssign(keys)  

}.optimizeReadOnlyMap()
* Removes the entry with the given [key] from this mutable map.
*  

@SinceKotlin("1.1")  

@kotlin.internal.InlineOnly  

public inline operator fun <K, V> MutableMap<K,  

V>.minusAssign(key: K) {  

    remove(key)  

}
* Removes all entries the keys of which are contained in  

the given [keys] collection from this mutable map.
*  

@SinceKotlin("1.1")  

@kotlin.internal.InlineOnly  

public inline operator fun <K, V> MutableMap<K,  

V>.minusAssign(keys: Iterable<K>) {  

    this.keys.removeAll(keys)  

}
* Removes all entries the keys of  

which are contained in the given [keys] array from this mutable map.
*  

@SinceKotlin("1.1")  

@kotlin.internal.InlineOnly  

public inline operator fun <K, V> MutableMap<K,  

V>.minusAssign(keys: Array<out K>) {  

    this.keys.removeAll(keys)  

}
* Removes all entries from the  

keys of which are contained in the given [keys] sequence from this mutable map.
*  

@SinceKotlin("1.1")  

@kotlin.internal.InlineOnly  

public inline operator fun <K, V> MutableMap<K,  

V>.minusAssign(keys: Sequence<K>) {  

    this.keys.removeAll(keys)  

}
// do not expose for now  

@PublishedApi  

internal fun <K, V> Map<K, V>.optimizeReadOnlyMap() = when (size) {  

    0 -> emptyMap()  

    1 -> toSingletonMapOrSelf()
} else -> this
}
* Copyright 2010-2021 JetBrains s.r.o. and Kotlin  

Programming Language contributors.
* Use of this source code is governed by the Apache 2.0 license that can be  

found in the license/LICENSE.txt file.
*  

@file:kotlin.jvm.JvmMultifileClass  

@file:kotlin.jvm.JvmName("SetsKt")  

@file:OptIn(kotlin.experimental  

.ExperimentalTypeInference::class)  

package kotlin.collections  

import kotlin.contracts.*  

internal object  

EmptySet : Set<Nothing>, Serializable {  

    private const val serialVersionUID: Long =  

3406603774387020532  

    override fun equals(other: Any?): Boolean = other is Set<*> && other.isEmpty()  

    override fun hashCode(): Int = 0  

    override fun toString(): String = "[]"  

    override val size: Int get() = 0  

    override fun isEmpty(): Boolean = true  

    override fun contains(element: Nothing): Boolean = false  

    override  

fun containsAll(elements: Collection<Nothing>): Boolean = elements.isEmpty()  

    override fun iterator():  

Iterator<Nothing> = EmptyIterator  

    private fun readResolve(): Any = EmptySet
}
* Returns an  

empty read-only set. The returned set is serializable (JVM).
* @sample  

samples.collections.Collections.Sets.emptyReadOnlySet
*  

public fun <T> emptySet(): Set<T> =  

EmptySet
* Returns a new read-only set with the given elements.
* Elements of the set are iterated in the

```

order they were specified.\n * The returned set is serializable (JVM).\n * @sample

```

samples.collections.Collections.Sets.readOnlySet\n * \npublic fun <T> setOf(vararg elements: T): Set<T> = if
(elements.size > 0) elements.toSet() else emptySet()\n\n/**\n * Returns an empty read-only set. The returned set is
serializable (JVM).\n * @sample samples.collections.Collections.Sets.emptyReadOnlySet\n
*\n*\n@kotlin.internal.InlineOnly\npublic inline fun <T> setOf(): Set<T> = emptySet()\n\n/**\n * Returns an empty
new [MutableSet].\n * \n * The returned set preserves the element iteration order.\n * @sample
samples.collections.Collections.Sets.emptyMutableSet\n
*\n*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <T> mutableSetOf(): MutableSet<T> =
LinkedHashSet()\n\n/**\n * Returns a new [MutableSet] with the given elements.\n * Elements of the set are
iterated in the order they were specified.\n * @sample samples.collections.Collections.Sets.mutableSet\n * \npublic
fun <T> mutableSetOf(vararg elements: T): MutableSet<T> =
elements.toCollection(LinkedHashSet(mapCapacity(elements.size)))\n\n/**\n * Returns an empty new [HashSet].
*\n*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <T> hashSetOf(): HashSet<T> =
HashSet()\n\n/**\n * Returns a new [HashSet] with the given elements. *\npublic fun <T> hashSetOf(vararg elements:
T): HashSet<T> = elements.toCollection(HashSet(mapCapacity(elements.size)))\n\n/**\n * Returns an empty new
[LinkedHashSet].\n * @sample samples.collections.Collections.Sets.emptyLinkedHashSet\n
*\n*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <T> linkedSetOf(): LinkedHashSet<T>
= LinkedHashSet()\n\n/**\n * Returns a new [LinkedHashSet] with the given elements.\n * Elements of the set are
iterated in the order they were specified.\n * @sample samples.collections.Collections.Sets.linkedHashSet\n
*\npublic fun <T> linkedSetOf(vararg elements: T): LinkedHashSet<T> =
elements.toCollection(LinkedHashSet(mapCapacity(elements.size)))\n\n/**\n * Returns a new read-only set either
with single given element, if it is not null, or empty set if the element is null.\n * The returned set is serializable
(JVM).\n * @sample samples.collections.Collections.Sets.setOfNotNull\n * \n*\n@SinceKotlin("1.4")\npublic fun <T
: Any> setOfNotNull(element: T?): Set<T> = if (element != null) setOf(element) else emptySet()\n\n/**\n * Returns
a new read-only set only with those given elements, that are not null.\n * Elements of the set are iterated in the order
they were specified.\n * The returned set is serializable (JVM).\n * @sample
samples.collections.Collections.Sets.setOfNotNull\n * \n*\n@SinceKotlin("1.4")\npublic fun <T : Any>
setOfNotNull(vararg elements: T?): Set<T> {\n    return elements.filterNotNullTo(LinkedHashSet())\n}\n\n/**\n *
Builds a new read-only [Set] by populating a [MutableSet] using the given [builderAction]\n * and returning a read-
only set with the same elements.\n * \n * The set passed as a receiver to the [builderAction] is valid only inside that
function.\n * Using it outside of the function produces an unspecified behavior.\n * \n * Elements of the set are
iterated in the order they were added by the [builderAction].\n * \n * The returned set is serializable (JVM).\n * \n
*\n * @sample samples.collections.Builders.Sets.buildSetSample\n
*\n*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\n@Su
ppress("DEPRECATION")\npublic inline fun <E> buildSet(@BuilderInference builderAction: MutableSet<E>().()
-> Unit): Set<E> {\n    contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n    return
buildSetInternal(builderAction)\n}\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\nintern
al expect inline fun <E> buildSetInternal(builderAction: MutableSet<E>().() -> Unit): Set<E>\n\n/**\n * Builds a
new read-only [Set] by populating a [MutableSet] using the given [builderAction]\n * and returning a read-only set
with the same elements.\n * \n * The set passed as a receiver to the [builderAction] is valid only inside that
function.\n * Using it outside of the function produces an unspecified behavior.\n * \n * [capacity] is used to hint the
expected number of elements added in the [builderAction].\n * \n * Elements of the set are iterated in the order they
were added by the [builderAction].\n * \n * The returned set is serializable (JVM).\n * \n * @throws
IllegalArgumentException if the given [capacity] is negative.\n * \n * @sample
samples.collections.Builders.Sets.buildSetSample\n
*\n*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\n@Su
ppress("DEPRECATION")\npublic inline fun <E> buildSet(capacity: Int, @BuilderInference builderAction:
MutableSet<E>().() -> Unit): Set<E> {\n    contract { callsInPlace(builderAction,

```



```

hashCode(): Int = \n    if (isEmpty()) -1 else (31 * (first xor (first ushr 32)) + (last xor (last ushr 32))).toInt()\n
override fun toString(): String = \"$first..$last\"\n\n companion object {\n    /** An empty range of values of
type Long. */\n    public val EMPTY: LongRange = LongRange(1, 0)\n } \n\n", /*\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("StringsKt")\n@file:Suppress("PLATFOR
M_CLASS_MAPPED_TO_KOTLIN")\n\npackage kotlin.text\n\n/**\n * Parses the string as a signed [Byte]
number and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n\n@SinceKotlin("1.1")\n\npublic fun String.toByteArrayOrNull(): Byte? = toByteOrNull(radix = 10)\n\n/**\n * Parses
the string as a signed [Byte] number and returns the result\n * or `null` if the string is not a valid representation of a
number.\n * *\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number
conversion.\n *\n\n@SinceKotlin("1.1")\n\npublic fun String.toByteArrayOrNull(radix: Int): Byte? {\n    val int =
this.toIntOrNull(radix) ?: return null\n    if (int < Byte.MIN_VALUE || int > Byte.MAX_VALUE) return null\n
return int.toByteArray()\n}\n\n/**\n * Parses the string as a [Short] number and returns the result\n * or `null` if the string
is not a valid representation of a number.\n *\n\n@SinceKotlin("1.1")\n\npublic fun String.toShortOrNull(): Short? =
toShortOrNull(radix = 10)\n\n/**\n * Parses the string as a [Short] number and returns the result\n * or `null` if the
string is not a valid representation of a number.\n * *\n * @throws IllegalArgumentException when [radix] is not a
valid radix for string to number conversion.\n *\n\n@SinceKotlin("1.1")\n\npublic fun String.toShortOrNull(radix:
Int): Short? {\n    val int = this.toIntOrNull(radix) ?: return null\n    if (int < Short.MIN_VALUE || int >
Short.MAX_VALUE) return null\n    return int.toShort()\n}\n\n/**\n * Parses the string as an [Int] number and
returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n\n@SinceKotlin("1.1")\n\npublic fun String.toIntOrNull(): Int? = toIntOrNull(radix = 10)\n\n/**\n * Parses the
string as an [Int] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n
*\n\n@SinceKotlin("1.1")\n\npublic fun String.toIntOrNull(radix: Int): Int? {\n    checkRadix(radix)\n\n    val length
= this.length\n    if (length == 0) return null\n\n    val start: Int\n    val isNegative: Boolean\n    val limit: Int\n    val
firstChar = this[0]\n    if (firstChar < '0') { // Possible leading sign\n        if (length == 1) return null // non-digit
(possible sign) only, no digits after\n\n        start = 1\n\n        if (firstChar == '-') {\n            isNegative = true\n
limit = Int.MIN_VALUE\n        } else if (firstChar == '+') {\n            isNegative = false\n            limit = -
Int.MAX_VALUE\n        } else\n            return null\n    } else {\n        start = 0\n        isNegative = false\n        limit
= -Int.MAX_VALUE\n    }\n\n    val limitForMaxRadix = (-Int.MAX_VALUE) / 36\n\n    var limitBeforeMul =
limitForMaxRadix\n    var result = 0\n    for (i in start until length) {\n        val digit = digitOf(this[i], radix)\n
if (digit < 0) return null\n        if (result < limitBeforeMul) {\n            if (limitBeforeMul == limitForMaxRadix) {\n
                limitBeforeMul = limit / radix\n\n                if (result < limitBeforeMul) {\n                    return null\n
                }\n            } else {\n                return null\n            }\n        }\n        result *= radix\n\n        if (result < limit + digit)
return null\n\n        result -= digit\n    }\n\n    return if (isNegative) result else -result\n}\n\n/**\n * Parses the string
as a [Long] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n\n@SinceKotlin("1.1")\n\npublic fun String.toLongOrNull(): Long? = toLongOrNull(radix = 10)\n\n/**\n * Parses
the string as a [Long] number and returns the result\n * or `null` if the string is not a valid representation of a
number.\n * *\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number
conversion.\n *\n\n@SinceKotlin("1.1")\n\npublic fun String.toLongOrNull(radix: Int): Long? {\n    checkRadix(radix)\n\n    val length = this.length\n    if (length == 0) return null\n\n    val start: Int\n    val isNegative:
Boolean\n    val limit: Long\n\n    val firstChar = this[0]\n    if (firstChar < '0') { // Possible leading sign\n        if
(length == 1) return null // non-digit (possible sign) only, no digits after\n\n        start = 1\n\n        if (firstChar == '-')
{\n            isNegative = true\n            limit = Long.MIN_VALUE\n        } else if (firstChar == '+') {\n           
isNegative = false\n            limit = -Long.MAX_VALUE\n        } else\n            return null\n    } else {\n        start =
0\n        isNegative = false\n        limit = -Long.MAX_VALUE\n    }\n\n    val limitForMaxRadix = (-
Long.MAX_VALUE) / 36\n\n    var limitBeforeMul = limitForMaxRadix\n    var result = 0L\n    for (i in start until

```

```

length) {\n    val digit = digitOf(this[i], radix)\n\n    if (digit < 0) return null\n    if (result < limitBeforeMul)\n    {\n        if (limitBeforeMul == limitForMaxRadix) {\n            limitBeforeMul = limit / radix\n\n            if\n            (result < limitBeforeMul) {\n                return null\n            }\n        } else {\n            return null\n        }\n    }\n\n    result *= radix\n\n    if (result < limit + digit) return null\n\n    result -= digit\n}\n\nreturn if (isNegative) result else -result\n}\n\n\ninternal fun numberFormatError(input: String): Nothing = throw\nNumberFormatException("Invalid number format: '$input')\n", "/*\n * Copyright 2010-2021 JetBrains s.r.o. and\n * Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that\n * can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.time\n\nimport kotlin.contracts.*\nimport\nkotlin.jvm.JvmInline\nimport kotlin.math.*\n\n/*\n * Represents the amount of time one instant of time is away\n * from another instant.\n * * A negative duration is possible in a situation when the second instant is earlier than the\n * first one.\n * * The type can store duration values up to \u00b1146 years with nanosecond precision,\n * and up to \u00b1146 million years with millisecond precision.\n * If a duration-returning operation provided in `kotlin.time`\n * produces a duration value that doesn't fit into the above range,\n * the returned `Duration` is infinite.\n * * An\n * infinite duration value [Duration.INFINITE] can be used to represent infinite timeouts.\n * * To construct a\n * duration use either the extension function [toDuration],\n * or the extension properties [hours], [minutes], [seconds],\n * and so on,\n * available on [Int], [Long], and [Double] numeric types.\n * * To get the value of this duration\n * expressed in a particular [duration units][DurationUnit]\n * use the functions [toInt], [toLong], and [toDouble]\n * or\n * the properties [inWholeHours], [inWholeMinutes], [inWholeSeconds], [inWholeNanoseconds], and so on.\n */\n\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalTime::class)\n@JvmInline\npublic value class\nDuration internal constructor(private val rawValue: Long) : Comparable<Duration> {\n\n    private val value: Long\n    get() = rawValue shr 1\n\n    private inline val unitDiscriminator: Int get() = rawValue.toInt() and 1\n\n    private fun\n    isInNanos() = unitDiscriminator == 0\n    private fun isInMillis() = unitDiscriminator == 1\n\n    private val\n    storageUnit get() = if (isInNanos()) DurationUnit.NANOSECONDS else DurationUnit.MILLISECONDS\n\n    init\n    {\n        if (durationAssertionsEnabled) {\n            if (isInNanos()) {\n                if (value !in -\n                MAX_NANOS..MAX_NANOS) throw AssertionError("$value ns is out of nanoseconds range")\n            } else\n            {\n                if (value !in -MAX_MILLIS..MAX_MILLIS) throw AssertionError("$value ms is out of milliseconds\n                range")\n                if (value in -MAX_NANOS_IN_MILLIS..MAX_NANOS_IN_MILLIS) throw\n                AssertionError("$value ms is denormalized")\n            }\n        }\n    }\n\n    companion object {\n        /** The\n        duration equal to exactly 0 seconds. */\n        public val ZERO: Duration = Duration(0L)\n\n        /** The duration\n        whose value is positive infinity. It is useful for representing timeouts that should never expire. */\n        public val\n        INFINITE: Duration = durationOfMillis(MAX_MILLIS)\n\n        internal val NEG_INFINITE: Duration =\n        durationOfMillis(-MAX_MILLIS)\n\n        /** Converts the given time duration [value] expressed in the specified\n        [sourceUnit] into the specified [targetUnit]. */\n        @ExperimentalTime\n        public fun convert(value: Double,\n        sourceUnit: DurationUnit, targetUnit: DurationUnit): Double =\n        convertDurationUnit(value, sourceUnit,\n        targetUnit)\n\n        // Duration construction extension properties in Duration companion scope\n\n        /** Returns a\n        [Duration] equal to this [Int] number of nanoseconds. */\n        @kotlin.internal.InlineOnly\n        public inline val\n        Int.nanoseconds get() = toDuration(DurationUnit.NANOSECONDS)\n\n        /** Returns a [Duration] equal to this\n        [Long] number of nanoseconds. */\n        @kotlin.internal.InlineOnly\n        public inline val Long.nanoseconds\n        get() = toDuration(DurationUnit.NANOSECONDS)\n\n        /**\n        * Returns a [Duration] equal to this\n        [Double] number of nanoseconds.\n        *\n        * Depending on its magnitude, the value is rounded to an integer\n        number of nanoseconds or milliseconds.\n        *\n        * @throws IllegalArgumentException if this [Double]\n        value is NaN. */\n        @kotlin.internal.InlineOnly\n        public inline val Double.nanoseconds get() =\n        toDuration(DurationUnit.NANOSECONDS)\n\n        /** Returns a [Duration] equal to this [Int] number of\n        microseconds. */\n        @kotlin.internal.InlineOnly\n        public inline val Int.microseconds get() =\n        toDuration(DurationUnit.MICROSECONDS)\n\n        /** Returns a [Duration] equal to this [Long] number of\n        microseconds. */\n        @kotlin.internal.InlineOnly\n        public inline val Long.microseconds get() =\n        toDuration(DurationUnit.MICROSECONDS)\n\n        /**\n        * Returns a [Duration] equal to this [Double]\n        number of microseconds.\n        *\n        * Depending on its magnitude, the value is rounded to an integer number

```



```

of nanoseconds or milliseconds.\n
    * @throws IllegalArgumentException if this [Double] value is
`NaN`.\n
    @kotlin.internal.InlineOnly\n
    public inline val Double.microseconds get() =
toDuration(DurationUnit.MICROSECONDS)\n\n
    /** Returns a [Duration] equal to this [Int] number of
milliseconds. */\n
    @kotlin.internal.InlineOnly\n
    public inline val Int.milliseconds get() =
toDuration(DurationUnit.MILLISECONDS)\n\n
    /** Returns a [Duration] equal to this [Long] number of
milliseconds. */\n
    @kotlin.internal.InlineOnly\n
    public inline val Long.milliseconds get() =
toDuration(DurationUnit.MILLISECONDS)\n\n
    /**\n
    * Returns a [Duration] equal to this [Double]
number of milliseconds.\n
    * Depending on its magnitude, the value is rounded to an integer number of
nanoseconds or milliseconds.\n
    * @throws IllegalArgumentException if this [Double] value is
`NaN`.\n
    * @kotlin.internal.InlineOnly\n
    public inline val Double.milliseconds get() =
toDuration(DurationUnit.MILLISECONDS)\n\n
    /** Returns a [Duration] equal to this [Int] number of
seconds. */\n
    @kotlin.internal.InlineOnly\n
    public inline val Int.seconds get() =
toDuration(DurationUnit.SECONDS)\n\n
    /** Returns a [Duration] equal to this [Long] number of seconds. */\n
    @kotlin.internal.InlineOnly\n
    public inline val Long.seconds get() =
toDuration(DurationUnit.SECONDS)\n\n
    /**\n
    * Returns a [Duration] equal to this [Double] number of
seconds.\n
    * Depending on its magnitude, the value is rounded to an integer number of nanoseconds or
milliseconds.\n
    * @throws IllegalArgumentException if this [Double] value is `NaN`.\n
    * @kotlin.internal.InlineOnly\n
    public inline val Double.seconds get() =
toDuration(DurationUnit.SECONDS)\n\n
    /** Returns a [Duration] equal to this [Int] number of minutes. */\n
    @kotlin.internal.InlineOnly\n
    public inline val Int.minutes get() = toDuration(DurationUnit.MINUTES)\n\n
    /** Returns a [Duration] equal to this [Long] number of minutes. */\n
    @kotlin.internal.InlineOnly\n
    public inline val Long.minutes get() = toDuration(DurationUnit.MINUTES)\n\n
    /**\n
    * Returns a
[Duration] equal to this [Double] number of minutes.\n
    * Depending on its magnitude, the value is
rounded to an integer number of nanoseconds or milliseconds.\n
    * @throws IllegalArgumentException
if this [Double] value is `NaN`.\n
    * @kotlin.internal.InlineOnly\n
    public inline val Double.minutes
get() = toDuration(DurationUnit.MINUTES)\n\n
    /** Returns a [Duration] equal to this [Int] number of hours.
*/\n
    @kotlin.internal.InlineOnly\n
    public inline val Int.hours get() = toDuration(DurationUnit.HOURS)\n\n
    /** Returns a [Duration] equal to this [Long] number of hours. */\n
    @kotlin.internal.InlineOnly\n
    public
inline val Long.hours get() = toDuration(DurationUnit.HOURS)\n\n
    /**\n
    * Returns a [Duration] equal to
this [Double] number of hours.\n
    * Depending on its magnitude, the value is rounded to an integer
number of nanoseconds or milliseconds.\n
    * @throws IllegalArgumentException if this [Double]
value is `NaN`.\n
    * @kotlin.internal.InlineOnly\n
    public inline val Double.hours get() =
toDuration(DurationUnit.HOURS)\n\n
    /** Returns a [Duration] equal to this [Int] number of days. */\n
    @kotlin.internal.InlineOnly\n
    public inline val Int.days get() = toDuration(DurationUnit.DAYS)\n\n
    /**
Returns a [Duration] equal to this [Long] number of days. */\n
    @kotlin.internal.InlineOnly\n
    public inline
val Long.days get() = toDuration(DurationUnit.DAYS)\n\n
    /**\n
    * Returns a [Duration] equal to this
[Double] number of days.\n
    * Depending on its magnitude, the value is rounded to an integer number
of nanoseconds or milliseconds.\n
    * @throws IllegalArgumentException if this [Double] value is
`NaN`.\n
    * @kotlin.internal.InlineOnly\n
    public inline val Double.days get() =
toDuration(DurationUnit.DAYS)\n\n
    // deprecated static factory functions\n\n
    /** Returns a [Duration]
representing the specified [value] number of nanoseconds. */\n
    @SinceKotlin("1.5")\n
    @ExperimentalTime\n
    @Deprecated("Use 'Int.nanoseconds' extension property from Duration.Companion
instead.", ReplaceWith("value.nanoseconds", "kotlin.time.Duration.Companion.nanoseconds"))\n
    @DeprecatedSinceKotlin(warningSince = "1.6")\n
    public fun nanoseconds(value: Int): Duration =
value.toDuration(DurationUnit.NANOSECONDS)\n\n
    /** Returns a [Duration] representing the specified
[value] number of nanoseconds. */\n
    @SinceKotlin("1.5")\n
    @ExperimentalTime\n
    @Deprecated("Use 'Long.nanoseconds' extension property from Duration.Companion instead.",
ReplaceWith("value.nanoseconds", "kotlin.time.Duration.Companion.nanoseconds"))\n

```

```

@DeprecatedSinceKotlin(warningSince = \"1.6\")\n    public fun nanoseconds(value: Long): Duration =
value.toDuration(DurationUnit.NANOSECONDS)\n\n    /**\n     * Returns a [Duration] representing the
specified [value] number of nanoseconds.\n     *\n     * @throws IllegalArgumentException if the provided
`Double` [value] is `NaN`.\n     */\n    @SinceKotlin(\"1.5\")\n    @ExperimentalTime\n@Deprecated(\"Use 'Double.nanoseconds' extension property from Duration.Companion instead.\",
ReplaceWith(\"value.nanoseconds\", \"kotlin.time.Duration.Companion.nanoseconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.6\")\n    public fun nanoseconds(value: Double): Duration =
value.toDuration(DurationUnit.NANOSECONDS)\n\n    /** Returns a [Duration] representing the specified
[value] number of microseconds. *\n     @SinceKotlin(\"1.5\")\n     @ExperimentalTime\n@Deprecated(\"Use 'Int.microseconds' extension property from Duration.Companion instead.\",
ReplaceWith(\"value.microseconds\", \"kotlin.time.Duration.Companion.microseconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.6\")\n    public fun microseconds(value: Int): Duration =
value.toDuration(DurationUnit.MICROSECONDS)\n\n    /** Returns a [Duration] representing the specified
[value] number of microseconds. *\n     @SinceKotlin(\"1.5\")\n     @ExperimentalTime\n@Deprecated(\"Use 'Long.microseconds' extension property from Duration.Companion instead.\",
ReplaceWith(\"value.microseconds\", \"kotlin.time.Duration.Companion.microseconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.6\")\n    public fun microseconds(value: Long): Duration =
value.toDuration(DurationUnit.MICROSECONDS)\n\n    /**\n     * Returns a [Duration] representing the
specified [value] number of microseconds.\n     *\n     * @throws IllegalArgumentException if the provided
`Double` [value] is `NaN`.\n     */\n    @SinceKotlin(\"1.5\")\n    @ExperimentalTime\n@Deprecated(\"Use 'Double.microseconds' extension property from Duration.Companion instead.\",
ReplaceWith(\"value.microseconds\", \"kotlin.time.Duration.Companion.microseconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.6\")\n    public fun microseconds(value: Double): Duration =
value.toDuration(DurationUnit.MICROSECONDS)\n\n    /** Returns a [Duration] representing the specified
[value] number of milliseconds. *\n     @SinceKotlin(\"1.5\")\n     @ExperimentalTime\n@Deprecated(\"Use 'Int.milliseconds' extension property from Duration.Companion instead.\",
ReplaceWith(\"value.milliseconds\", \"kotlin.time.Duration.Companion.milliseconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.6\")\n    public fun milliseconds(value: Int): Duration =
value.toDuration(DurationUnit.MILLISECONDS)\n\n    /** Returns a [Duration] representing the specified
[value] number of milliseconds. *\n     @SinceKotlin(\"1.5\")\n     @ExperimentalTime\n@Deprecated(\"Use 'Long.milliseconds' extension property from Duration.Companion instead.\",
ReplaceWith(\"value.milliseconds\", \"kotlin.time.Duration.Companion.milliseconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.6\")\n    public fun milliseconds(value: Long): Duration =
value.toDuration(DurationUnit.MILLISECONDS)\n\n    /**\n     * Returns a [Duration] representing the
specified [value] number of milliseconds.\n     *\n     * @throws IllegalArgumentException if the provided
`Double` [value] is `NaN`.\n     */\n    @SinceKotlin(\"1.5\")\n    @ExperimentalTime\n@Deprecated(\"Use 'Double.milliseconds' extension property from Duration.Companion instead.\",
ReplaceWith(\"value.milliseconds\", \"kotlin.time.Duration.Companion.milliseconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.6\")\n    public fun milliseconds(value: Double): Duration =
value.toDuration(DurationUnit.MILLISECONDS)\n\n    /** Returns a [Duration] representing the specified
[value] number of seconds. *\n     @SinceKotlin(\"1.5\")\n     @ExperimentalTime\n    @Deprecated(\"Use
'Int.seconds' extension property from Duration.Companion instead.\", ReplaceWith(\"value.seconds\",
\"kotlin.time.Duration.Companion.seconds\"))\n    @DeprecatedSinceKotlin(warningSince = \"1.6\")\n    public
fun seconds(value: Int): Duration = value.toDuration(DurationUnit.SECONDS)\n\n    /** Returns a [Duration]
representing the specified [value] number of seconds. *\n     @SinceKotlin(\"1.5\")\n     @ExperimentalTime\n    @Deprecated(\"Use 'Long.seconds' extension property from Duration.Companion instead.\",
ReplaceWith(\"value.seconds\", \"kotlin.time.Duration.Companion.seconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.6\")\n    public fun seconds(value: Long): Duration =

```

```

value.toDuration(DurationUnit.SECONDS)\n\n    /**\n    * Returns a [Duration] representing the specified
[value] number of seconds.\n    *\n    * @throws IllegalArgumentException if the provided `Double` [value] is
`NaN`.\n    *\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n    @Deprecated("Use
'Double.seconds' extension property from Duration.Companion instead.", ReplaceWith("value.seconds",
"\"kotlin.time.Duration.Companion.seconds\""))\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public
fun seconds(value: Double): Duration = value.toDuration(DurationUnit.SECONDS)\n\n    /** Returns a
[Duration] representing the specified [value] number of minutes. *\n    @SinceKotlin("1.5")\n
@ExperimentalTime\n    @Deprecated("Use 'Int.minutes' extension property from Duration.Companion
instead.", ReplaceWith("value.minutes", "\"kotlin.time.Duration.Companion.minutes\""))\n
@DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun minutes(value: Int): Duration =
value.toDuration(DurationUnit.MINUTES)\n\n    /** Returns a [Duration] representing the specified [value]
number of minutes. *\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n    @Deprecated("Use
'Long.minutes' extension property from Duration.Companion instead.", ReplaceWith("value.minutes",
"\"kotlin.time.Duration.Companion.minutes\""))\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public
fun minutes(value: Long): Duration = value.toDuration(DurationUnit.MINUTES)\n\n    /**\n    * Returns a
[Duration] representing the specified [value] number of minutes.\n    *\n    * @throws
IllegalArgumentException if the provided `Double` [value] is `NaN`.\n    *\n    @SinceKotlin("1.5")\n
@ExperimentalTime\n    @Deprecated("Use 'Double.minutes' extension property from Duration.Companion
instead.", ReplaceWith("value.minutes", "\"kotlin.time.Duration.Companion.minutes\""))\n
@DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun minutes(value: Double): Duration =
value.toDuration(DurationUnit.MINUTES)\n\n    /** Returns a [Duration] representing the specified [value]
number of hours. *\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n    @Deprecated("Use 'Int.hours'
extension property from Duration.Companion instead.", ReplaceWith("value.hours",
"\"kotlin.time.Duration.Companion.hours\""))\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public
fun hours(value: Int): Duration = value.toDuration(DurationUnit.HOURS)\n\n    /** Returns a [Duration]
representing the specified [value] number of hours. *\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n
@Deprecated("Use 'Long.hours' extension property from Duration.Companion instead.",
ReplaceWith("value.hours", "\"kotlin.time.Duration.Companion.hours\""))\n
@DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun hours(value: Long): Duration =
value.toDuration(DurationUnit.HOURS)\n\n    /**\n    * Returns a [Duration] representing the specified
[value] number of hours.\n    *\n    * @throws IllegalArgumentException if the provided `Double` [value] is
`NaN`.\n    *\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n    @Deprecated("Use 'Double.hours'
extension property from Duration.Companion instead.", ReplaceWith("value.hours",
"\"kotlin.time.Duration.Companion.hours\""))\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public
fun hours(value: Double): Duration = value.toDuration(DurationUnit.HOURS)\n\n    /** Returns a [Duration]
representing the specified [value] number of days. *\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n
@Deprecated("Use 'Int.days' extension property from Duration.Companion instead.", ReplaceWith("value.days",
"\"kotlin.time.Duration.Companion.days\""))\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public
fun days(value: Int): Duration = value.toDuration(DurationUnit.DAYS)\n\n    /** Returns a [Duration]
representing the specified [value] number of days. *\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n
@Deprecated("Use 'Long.days' extension property from Duration.Companion instead.",
ReplaceWith("value.days", "\"kotlin.time.Duration.Companion.days\""))\n
@DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun days(value: Long): Duration =
value.toDuration(DurationUnit.DAYS)\n\n    /**\n    * Returns a [Duration] representing the specified [value]
number of days.\n    *\n    * @throws IllegalArgumentException if the provided `Double` [value] is `NaN`.\n
*\n    *\n    *\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n    @Deprecated("Use 'Double.days' extension
property from Duration.Companion instead.", ReplaceWith("value.days",
"\"kotlin.time.Duration.Companion.days\""))\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public

```

```

fun days(value: Double): Duration = value.toDuration(DurationUnit.DAYS)\n\n    /**\n     * Parses a string that
represents a duration and returns the parsed [Duration] value.\n     *\n     * The following formats are
accepted:\n     *\n     * - ISO-8601 Duration format, e.g. `P1DT2H3M4.058S`, see [toIsoString] and
[parseIsoString].\n     * - The format of string returned by the default [Duration.toString] and `toString` in a
specific unit,\n     * e.g. `10s`, `1h 30m` or `-(1h 30m)`.\n     *\n     * @throws IllegalArgumentException if
the string doesn't represent a duration in any of the supported formats.\n     * @sample
samples.time.Durations.parse\n     */\n    public fun parse(value: String): Duration = try {\n
parseDuration(value, strictIso = false)\n    } catch (e: IllegalArgumentException) {\n        throw
IllegalArgumentException("Invalid duration string format: '$value'.", e)\n    }\n\n    /**\n     * Parses a
string that represents a duration in a restricted ISO-8601 composite representation\n     * and returns the parsed
[Duration] value.\n     * Composite representation is a relaxed version of ISO-8601 duration format that
supports\n     * negative durations and negative values of individual components.\n     *\n     * The following
restrictions are imposed:\n     *\n     * - The only allowed non-time designator is days (`D`), `Y` (years), `W`
(weeks), and `M` (months) are not supported.\n     * - Day is considered to be exactly 24 hours (24-hour clock
time scale).\n     * - Alternative week-based representation `[P][number][W]` is not supported.\n     *\n     *
@throws IllegalArgumentException if the string doesn't represent a duration in ISO-8601 format.\n     *
@sample samples.time.Durations.parseIsoString\n     */\n    public fun parseIsoString(value: String): Duration
= try {\n        parseDuration(value, strictIso = true)\n    } catch (e: IllegalArgumentException) {\n        throw
IllegalArgumentException("Invalid ISO duration string format: '$value'.", e)\n    }\n\n    /**\n     * Parses a
string that represents a duration and returns the parsed [Duration] value,\n     * or `null` if the string doesn't
represent a duration in any of the supported formats.\n     *\n     * The following formats are accepted:\n     *\n     *
- Restricted ISO-8601 duration composite representation, e.g. `P1DT2H3M4.058S`, see [toIsoString] and
[parseIsoString].\n     * - The format of string returned by the default [Duration.toString] and `toString` in a
specific unit,\n     * e.g. `10s`, `1h 30m` or `-(1h 30m)`.\n     * @sample samples.time.Durations.parse\n     */\n    public fun parseOrNull(value: String): Duration? = try {\n
parseDuration(value, strictIso = false)\n    } catch (e: IllegalArgumentException) {\n        null\n    }\n\n    /**\n     * Parses a string that represents a
duration in restricted ISO-8601 composite representation\n     * and returns the parsed [Duration] value or `null` if
the string doesn't represent a duration in the format\n     * acceptable by [parseIsoString].\n     *\n     *
@sample samples.time.Durations.parseIsoString\n     */\n    public fun parseIsoStringOrNull(value: String):
Duration? = try {\n        parseDuration(value, strictIso = true)\n    } catch (e: IllegalArgumentException) {\n
null\n    }\n\n    /**\n     * Returns the negative of this value.\n     */\n    public operator
fun unaryMinus(): Duration = durationOf(-value, unitDiscriminator)\n\n    /**\n     * Returns a duration whose value
is the sum of this and [other] duration values.\n     *\n     * @throws IllegalArgumentException if the operation
results in an undefined value for the given arguments,\n     * e.g. when adding infinite durations of different sign.\n     */\n    public operator fun plus(other: Duration): Duration {\n
when {\n        this.isInfinite() -> {\n            if
(other.isFinite() || (this.rawValue xor other.rawValue >= 0))\n                return this\n            else\n
throw IllegalArgumentException("Summing infinite durations of different signs yields an undefined result.")\n        }\n        other.isInfinite() -> return other\n    }\n    return when {\n        this.unitDiscriminator ==
other.unitDiscriminator -> {\n            val result = this.value + other.value // never overflows long, but can
overflow long63\n            when {\n                isInNanos() ->\n
durationOfNanosNormalized(result)\n            else ->\n                durationOfMillisNormalized(result)\n            }\n        }\n        this.isInMillis() ->\n            addValuesMixedRanges(this.value, other.value)\n    }\n    else ->\n        addValuesMixedRanges(other.value, this.value)\n    }\n}\n\n    private fun
addValuesMixedRanges(thisMillis: Long, otherNanos: Long): Duration {\n        val otherMillis =
nanosToMillis(otherNanos)\n        val resultMillis = thisMillis + otherMillis\n        return if (resultMillis in -
MAX_NANOS_IN_MILLIS..MAX_NANOS_IN_MILLIS) {\n            val otherNanoRemainder = otherNanos -
millisToNanos(otherMillis)\n            durationOfNanos(millisToNanos(resultMillis) + otherNanoRemainder)\n        }\n    } else {\n        durationOfMillis(resultMillis.coerceIn(-MAX_MILLIS, MAX_MILLIS))\n    }\n}\n\n    /**\n

```

```

* Returns a duration whose value is the difference between this and [other] duration values.\n
*\n
* @throws
IllegalArgumentException if the operation results in an undefined value for the given arguments,\n
* e.g. when
subtracting infinite durations of the same sign.\n
*/\n
public operator fun minus(other: Duration): Duration =
this + (-other)\n\n
/**\n
* Returns a duration whose value is this duration value multiplied by the given [scale]
number.\n
*\n
* @throws ArgumentException if the operation results in an undefined value for the given
arguments,\n
* e.g. when multiplying an infinite duration by zero.\n
*/\n
public operator fun times(scale: Int):
Duration {\n
    if (isInfinite()) {\n
        return when {\n
            scale == 0 -> throw
IllegalArgumentException("Multiplying infinite duration by zero yields an undefined result.")\n
            scale > 0
-> this\n
            else -> -this\n
        }\n
    }\n
    if (scale == 0) return ZERO\n\n
    val value = value\n
    val result = value * scale\n
    return if (isInNanos()) {\n
        if (value in (MAX_NANOS /
Int.MIN_VALUE)..(-MAX_NANOS / Int.MIN_VALUE)) {\n
            // can't overflow nanos range for any
scale\n
            durationOfNanos(result)\n
        } else {\n
            if (result / scale == value) {\n
durationOfNanosNormalized(result)\n
            } else {\n
                val millis = nanosToMillis(value)\n
                val remNanos = value - millisToNanos(millis)\n
                val resultMillis = millis * scale\n
                val
totalMillis = resultMillis + nanosToMillis(remNanos * scale)\n
                if (resultMillis / scale == millis &&
totalMillis xor resultMillis >= 0) {\n
                    durationOfMillis(totalMillis.coerceIn(-
MAX_MILLIS..MAX_MILLIS))\n
                } else {\n
                    if (value.sign * scale.sign > 0) INFINITE
else NEG_INFINITE\n
                }\n
            }\n
        } else {\n
            if (result / scale == value) {\n
                durationOfMillis(result.coerceIn(-MAX_MILLIS..MAX_MILLIS))\n
            } else {\n
                if (value.sign
* scale.sign > 0) INFINITE else NEG_INFINITE\n
            }\n
        }\n
    }\n\n
/**\n
* Returns a duration whose
value is this duration value multiplied by the given [scale] number.\n
*\n
* The operation may involve rounding
when the result cannot be represented exactly with a [Double] number.\n
*\n
* @throws
IllegalArgumentException if the operation results in an undefined value for the given arguments,\n
* e.g. when
multiplying an infinite duration by zero.\n
*/\n
public operator fun times(scale: Double): Duration {\n
    val
intScale = scale.roundToInt()\n
    if (intScale.toDouble() == scale) {\n
        return times(intScale)\n
    }\n\n
    val unit = storageUnit\n
    val result = toDouble(unit) * scale\n
    return result.toDuration(unit)\n
}\n\n
/**\n
* Returns a duration whose value is this duration value divided by the given [scale] number.\n
*\n
*
@throws ArgumentException if the operation results in an undefined value for the given arguments,\n
*
e.g. when dividing zero duration by zero.\n
*/\n
public operator fun div(scale: Int): Duration {\n
    if (scale ==
0) {\n
        return when {\n
            isPositive() -> INFINITE\n
            isNegative() -> NEG_INFINITE\n
            else -> throw ArgumentException("Dividing zero duration by zero yields an undefined result.")\n
        }\n
    }\n
    if (isInNanos()) {\n
        return durationOfNanos(value / scale)\n
    } else {\n
        if
(isInfinite())\n
            return this * scale.sign\n\n
        val result = value / scale\n\n
        if (result in -
MAX_NANOS_IN_MILLIS..MAX_NANOS_IN_MILLIS) {\n
            val rem = millisToNanos(value - (result *
scale)) / scale\n
            return durationOfNanos(millisToNanos(result) + rem)\n
        }\n
        return
durationOfMillis(result)\n
    }\n
}\n\n
/**\n
* Returns a duration whose value is this duration value divided
by the given [scale] number.\n
*\n
* @throws ArgumentException if the operation results in an
undefined value for the given arguments,\n
* e.g. when dividing an infinite duration by infinity or zero duration
by zero.\n
*/\n
public operator fun div(scale: Double): Duration {\n
    val intScale = scale.roundToInt()\n
    if (intScale.toDouble() == scale && intScale != 0) {\n
        return div(intScale)\n
    }\n\n
    val unit =
storageUnit\n
    val result = toDouble(unit) / scale\n
    return result.toDuration(unit)\n
}\n\n
/** Returns a
number that is the ratio of this and [other] duration values. */\n
public operator fun div(other: Duration): Double
{\n
    val coarserUnit = maxOf(this.storageUnit, other.storageUnit)\n
    return this.toDouble(coarserUnit) /
other.toDouble(coarserUnit)\n
}\n\n
/** Returns true, if the duration value is less than zero. */\n
public fun
isNegative(): Boolean = rawValue < 0\n\n
/** Returns true, if the duration value is greater than zero. */\n
public fun
isPositive(): Boolean = rawValue > 0\n\n
/** Returns true, if the duration value is infinite. */\n
public fun
isInfinite(): Boolean = rawValue == INFINITE.rawValue || rawValue == NEG_INFINITE.rawValue\n\n
/**
Returns true, if the duration value is finite. */\n
public fun isFinite(): Boolean = !isInfinite()\n\n
/** Returns the

```

```

absolute value of this value. The returned value is always non-negative. */
public val absoluteValue: Duration
get() = if (isNegative()) -this else this
override fun compareTo(other: Duration): Int {
    val compareBits = this.rawValue xor other.rawValue
    if (compareBits < 0 || compareBits.toInt() and 1 == 0) // different signs or
    same sign/same range
        return this.rawValue.compareTo(other.rawValue) // same sign/different
    ranges
    val r = this.unitDiscriminator - other.unitDiscriminator // compare ranges
    return if (isNegative())
    -r else r
}
/**
 * Splits this duration into days, hours, minutes,
    seconds, and nanoseconds and executes the given [action] with these components.
    * The result of [action] is
    returned as the result of this function.
    * - `nanoseconds` represents the whole number of nanoseconds in
    this duration, and its absolute value is less than 1_000_000_000;
    * - `seconds` represents the whole number of
    seconds in this duration, and its absolute value is less than 60;
    * - `minutes` represents the whole number of
    minutes in this duration, and its absolute value is less than 60;
    * - `hours` represents the whole number of hours
    in this duration, and its absolute value is less than 24;
    * - `days` represents the whole number of days in this
    duration.
    * Infinite durations are represented as either [Long.MAX_VALUE] days, or
    [Long.MIN_VALUE] days (depending on the sign of infinity),
    * and zeroes in the lower components.
    */
public inline fun <T> toComponents(action: (days: Long, hours: Int, minutes: Int, seconds: Int, nanoseconds: Int) -
> T): T {
    contract { callsInPlace(action, InvocationKind.EXACTLY_ONCE) }
    return
    action(inWholeDays, hoursComponent, minutesComponent, secondsComponent, nanosecondsComponent)
}
/**
 * Splits this duration into hours, minutes, seconds, and nanoseconds and executes the given [action] with
    these components.
    * The result of [action] is returned as the result of this function.
    * - `nanoseconds`
    represents the whole number of nanoseconds in this duration, and its absolute value is less than 1_000_000_000;
    * - `seconds` represents the whole number of seconds in this duration, and its absolute value is less than 60;
    * -
    `minutes` represents the whole number of minutes in this duration, and its absolute value is less than 60;
    * -
    `hours` represents the whole number of hours in this duration.
    * Infinite durations are represented as
    either [Long.MAX_VALUE] hours, or [Long.MIN_VALUE] hours (depending on the sign of infinity),
    * and
    zeroes in the lower components.
    */
public inline fun <T> toComponents(action: (hours: Long, minutes: Int,
seconds: Int, nanoseconds: Int) -> T): T {
    contract { callsInPlace(action, InvocationKind.EXACTLY_ONCE)
}
    return action(inWholeHours, minutesComponent, secondsComponent, nanosecondsComponent)
}
/**
 * Splits this duration into minutes, seconds, and nanoseconds and executes the given [action] with these
    components.
    * The result of [action] is returned as the result of this function.
    * - `nanoseconds`
    represents the whole number of nanoseconds in this duration, and its absolute value is less than 1_000_000_000;
    * - `seconds` represents the whole number of seconds in this duration, and its absolute value is less than 60;
    * -
    `minutes` represents the whole number of minutes in this duration.
    * Infinite durations are represented as
    either [Long.MAX_VALUE] minutes, or [Long.MIN_VALUE] minutes (depending on the sign of infinity),
    *
    and zeroes in the lower components.
    */
public inline fun <T> toComponents(action: (minutes: Long,
seconds: Int, nanoseconds: Int) -> T): T {
    contract { callsInPlace(action, InvocationKind.EXACTLY_ONCE)
}
    return action(inWholeMinutes, secondsComponent, nanosecondsComponent)
}
/**
 * Splits
    this duration into seconds, and nanoseconds and executes the given [action] with these components.
    * The result
    of [action] is returned as the result of this function.
    * - `nanoseconds` represents the whole number of
    nanoseconds in this duration, and its absolute value is less than 1_000_000_000;
    * - `seconds` represents the
    whole number of seconds in this duration.
    * Infinite durations are represented as either
    [Long.MAX_VALUE] seconds, or [Long.MIN_VALUE] seconds (depending on the sign of infinity),
    * and
    zero nanoseconds.
    */
public inline fun <T> toComponents(action: (seconds: Long, nanoseconds: Int) -> T):
T {
    contract { callsInPlace(action, InvocationKind.EXACTLY_ONCE) }
    return
    action(inWholeSeconds, nanosecondsComponent)
}
@PublishedApi
internal val hoursComponent:
Int
get() = if (isInfinite()) 0 else (inWholeHours % 24).toInt()
@PublishedApi
internal val
minutesComponent: Int
get() = if (isInfinite()) 0 else (inWholeMinutes % 60).toInt()
@PublishedApi
internal val secondsComponent: Int
get() = if (isInfinite()) 0 else (inWholeSeconds % 60).toInt()
@PublishedApi
internal val nanosecondsComponent: Int
get() = when {
    isInfinite() -> 0
}

```

```

isInMillis() -> millisToNanos(value % 1_000).toInt()\n      else -> (value % 1_000_000_000).toInt()\n
}\n\n // conversion to units\n\n /**\n * Returns the value of this duration expressed as a [Double] number of
the specified [unit].\n *\n * The operation may involve rounding when the result cannot be represented exactly
with a [Double] number.\n *\n * An infinite duration value is converted either to
[Double.POSITIVE_INFINITY] or [Double.NEGATIVE_INFINITY] depending on its sign.\n */\n public fun
toDouble(unit: DurationUnit): Double {\n      return when (rawValue) {\n          INFINITE.rawValue ->
Double.POSITIVE_INFINITY\n          NEG_INFIMATE.rawValue -> Double.NEGATIVE_INFINITY\n
else -> {\n              // TODO: whether it's ok to convert to Double before scaling\n
convertDurationUnit(value.toDouble(), storageUnit, unit)\n          }\n      }\n\n /**\n * Returns the value
of this duration expressed as a [Long] number of the specified [unit].\n *\n * If the result doesn't fit in the range
of [Long] type, it is coerced into that range:\n * - [Long.MIN_VALUE] is returned if it's less than
`Long.MIN_VALUE`,\n * - [Long.MAX_VALUE] is returned if it's greater than `Long.MAX_VALUE`.\n *\n
* An infinite duration value is converted either to [Long.MAX_VALUE] or [Long.MIN_VALUE] depending on
its sign.\n */\n public fun toLong(unit: DurationUnit): Long {\n      return when (rawValue) {\n
INFINITE.rawValue -> Long.MAX_VALUE\n          NEG_INFIMATE.rawValue -> Long.MIN_VALUE\n
else -> convertDurationUnit(value, storageUnit, unit)\n      }\n\n /**\n * Returns the value of this
duration expressed as an [Int] number of the specified [unit].\n *\n * If the result doesn't fit in the range of [Int]
type, it is coerced into that range:\n * - [Int.MIN_VALUE] is returned if it's less than `Int.MIN_VALUE`,\n * -
[Int.MAX_VALUE] is returned if it's greater than `Int.MAX_VALUE`.\n *\n * An infinite duration value is
converted either to [Int.MAX_VALUE] or [Int.MIN_VALUE] depending on its sign.\n */\n public fun
toInt(unit: DurationUnit): Int =\n      toLong(unit).coerceIn(Int.MIN_VALUE.toLong(),
Int.MAX_VALUE.toLong()).toInt()\n\n /** The value of this duration expressed as a [Double] number of days.
*\n @ExperimentalTime\n @Deprecated("Use inWholeDays property instead or convert toDouble(DAYS) if a
double value is required.", ReplaceWith("toDouble(DurationUnit.DAYS)"))\n public val inDays: Double get() =
toDouble(DurationUnit.DAYS)\n\n /** The value of this duration expressed as a [Double] number of hours. *\n
@ExperimentalTime\n @Deprecated("Use inWholeHours property instead or convert toDouble(HOURS) if a
double value is required.", ReplaceWith("toDouble(DurationUnit.HOURS)"))\n public val inHours: Double
get() = toDouble(DurationUnit.HOURS)\n\n /** The value of this duration expressed as a [Double] number of
minutes. *\n @ExperimentalTime\n @Deprecated("Use inWholeMinutes property instead or convert
toDouble(MINUTES) if a double value is required.", ReplaceWith("toDouble(DurationUnit.MINUTES)"))\n
public val inMinutes: Double get() = toDouble(DurationUnit.MINUTES)\n\n /** The value of this duration
expressed as a [Double] number of seconds. *\n @ExperimentalTime\n @Deprecated("Use inWholeSeconds
property instead or convert toDouble(SECONDS) if a double value is required.",
ReplaceWith("toDouble(DurationUnit.SECONDS)"))\n public val inSeconds: Double get() =
toDouble(DurationUnit.SECONDS)\n\n /** The value of this duration expressed as a [Double] number of
milliseconds. *\n @ExperimentalTime\n @Deprecated("Use inWholeMilliseconds property instead or convert
toDouble(MILLISECONDS) if a double value is required.",
ReplaceWith("toDouble(DurationUnit.MILLISECONDS)"))\n public val inMilliseconds: Double get() =
toDouble(DurationUnit.MILLISECONDS)\n\n /** The value of this duration expressed as a [Double] number of
microseconds. *\n @ExperimentalTime\n @Deprecated("Use inWholeMicroseconds property instead or
convert toDouble(MICROSECONDS) if a double value is required.",
ReplaceWith("toDouble(DurationUnit.MICROSECONDS)"))\n public val inMicroseconds: Double get() =
toDouble(DurationUnit.MICROSECONDS)\n\n /** The value of this duration expressed as a [Double] number of
nanoseconds. *\n @ExperimentalTime\n @Deprecated("Use inWholeNanoseconds property instead or convert
toDouble(NANOSECONDS) if a double value is required.",
ReplaceWith("toDouble(DurationUnit.NANOSECONDS)"))\n public val inNanoseconds: Double get() =
toDouble(DurationUnit.NANOSECONDS)\n\n /**\n * The value of this duration expressed as a [Long]
number of days.\n *\n * An infinite duration value is converted either to [Long.MAX_VALUE] or

```

```

[Long.MIN_VALUE] depending on its sign.\n
 */\n
 public val inWholeDays: Long\n
     get() =
toLong(DurationUnit.DAYS)\n\n
 /**\n
  * The value of this duration expressed as a [Long] number of hours.\n
  *\n
  * An infinite duration value is converted either to [Long.MAX_VALUE] or [Long.MIN_VALUE] depending
on its sign.\n
 */\n
 public val inWholeHours: Long\n
     get() = toLong(DurationUnit.HOURS)\n\n
 /**\n
  * The value of this duration expressed as a [Long] number of minutes.\n
  *\n
  * An infinite duration value is
converted either to [Long.MAX_VALUE] or [Long.MIN_VALUE] depending on its sign.\n
 */\n
 public val
inWholeMinutes: Long\n
     get() = toLong(DurationUnit.MINUTES)\n\n
 /**\n
  * The value of this duration
expressed as a [Long] number of seconds.\n
  *\n
  * An infinite duration value is converted either to
[Long.MAX_VALUE] or [Long.MIN_VALUE] depending on its sign.\n
 */\n
 public val inWholeSeconds:
Long\n
     get() = toLong(DurationUnit.SECONDS)\n\n
 /**\n
  * The value of this duration expressed as a
[Long] number of milliseconds.\n
  *\n
  * An infinite duration value is converted either to [Long.MAX_VALUE]
or [Long.MIN_VALUE] depending on its sign.\n
 */\n
 public val inWholeMilliseconds: Long\n
     get() {\n
     return if (isInMillis() && isFinite()) value else toLong(DurationUnit.MILLISECONDS)\n
     }\n\n
 /**\n
  * The value of this duration expressed as a [Long] number of microseconds.\n
  *\n
  * If the result doesn't fit in the
range of [Long] type, it is coerced into that range:\n
  * - [Long.MIN_VALUE] is returned if it's less than
`Long.MIN_VALUE`,\n
  * - [Long.MAX_VALUE] is returned if it's greater than `Long.MAX_VALUE`.\n
  *\n
  * An infinite duration value is converted either to [Long.MAX_VALUE] or [Long.MIN_VALUE] depending on
its sign.\n
 */\n
 public val inWholeMicroseconds: Long\n
     get() =
toLong(DurationUnit.MICROSECONDS)\n\n
 /**\n
  * The value of this duration expressed as a [Long] number
of nanoseconds.\n
  *\n
  * If the result doesn't fit in the range of [Long] type, it is coerced into that range:\n
  * -
[Long.MIN_VALUE] is returned if it's less than `Long.MIN_VALUE`,\n
  * - [Long.MAX_VALUE] is returned if
it's greater than `Long.MAX_VALUE`.\n
  *\n
  * An infinite duration value is converted either to
[Long.MAX_VALUE] or [Long.MIN_VALUE] depending on its sign.\n
 */\n
 public val inWholeNanoseconds:
Long\n
     get() {\n
     val value = value\n
     return when {\n
         isInNanos() -> value\n
         value > Long.MAX_VALUE / NANOS_IN_MILLIS -> Long.MAX_VALUE\n
         value <
Long.MIN_VALUE / NANOS_IN_MILLIS -> Long.MIN_VALUE\n
     } else -> millisToNanos(value)\n
     }\n
     }\n\n
 // shortcuts\n\n
 /**\n
  * Returns the value of this duration expressed as a [Long] number of
nanoseconds.\n
  *\n
  * If the value doesn't fit in the range of [Long] type, it is coerced into that range, see the
conversion [Double.toLong] for details.\n
  *\n
  * The range of durations that can be expressed as a `Long`
number of nanoseconds is approximately \u00b1292 years.\n
  */\n
 @ExperimentalTime\n
 @Deprecated("Use
inWholeNanoseconds property instead.", ReplaceWith("this.inWholeNanoseconds"))\n
 public fun
toLongNanoseconds(): Long = inWholeNanoseconds\n\n
 /**\n
  * Returns the value of this duration expressed as
a [Long] number of milliseconds.\n
  *\n
  * The value is coerced to the range of [Long] type, if it doesn't fit in
that range, see the conversion [Double.toLong] for details.\n
  *\n
  * The range of durations that can be expressed
as a `Long` number of milliseconds is approximately \u00b1292 million years.\n
  */\n
 @ExperimentalTime\n
 @Deprecated("Use inWholeMilliseconds property instead.", ReplaceWith("this.inWholeMilliseconds"))\n
 public fun toLongMilliseconds(): Long = inWholeMilliseconds\n\n
 /**\n
  * Returns a string representation of
this duration value\n
  * expressed as a combination of numeric components, each in its own unit.\n
  *\n
  * Each
component is a number followed by the unit abbreviated name: `d`, `h`, `m`, `s`:\n
  * * `5h`, `1d 12h`, `1h 0m
30.340s`.\n
  * The last component, usually seconds, can be a number with a fractional part.\n
  *\n
  * If the
duration is less than a second, it is represented as a single number\n
  * with one of sub-second units: `ms`
(milliseconds), `us` (microseconds), or `ns` (nanoseconds):\n
  * * `140.884ms`, `500us`, `24ns`.\n
  *\n
  * A
negative duration is prefixed with `-` sign and, if it consists of multiple components, surrounded with parentheses:\n
  * * `-12m` and `-(1h 30m)`.\n
  *\n
  * Special cases:\n
  * * - an infinite duration is formatted as `"Infinity"` or
`"-Infinity"` without a unit.\n
  *\n
  * It's recommended to use [toIsoString] that uses more strict ISO-8601
format instead of this `toString`\n
  * when you want to convert a duration to a string in cases of serialization,
interchange, etc.\n
  *\n
  * @sample samples.time.Durations.toStringDefault\n
  */\n
 override fun toString():
String = when (rawValue) {\n
     0L -> "0s"\n
     INFINITE.rawValue -> "Infinity"\n

```



```

*^@SinceKotlin("1.6")^@WasExperimental(ExperimentalTime::class)^npublic fun Int.toDuration(unit:
DurationUnit): Duration {^n    return if (unit <= DurationUnit.SECONDS) {^n
durationOfNanos(convertDurationUnitOverflow(this.toLong(), unit, DurationUnit.NANOSECONDS))^n    } else^n
    toLong().toDuration(unit)^n }^n/^n/** Returns a [Duration] equal to this [Long] number of the specified [unit].
*^@SinceKotlin("1.6")^@WasExperimental(ExperimentalTime::class)^npublic fun Long.toDuration(unit:
DurationUnit): Duration {^n    val maxNsInUnit = convertDurationUnitOverflow(MAX_NANOS,
DurationUnit.NANOSECONDS, unit)^n    if (this in -maxNsInUnit..maxNsInUnit) {^n        return
durationOfNanos(convertDurationUnitOverflow(this, unit, DurationUnit.NANOSECONDS))^n    } else {^n        val
millis = convertDurationUnit(this, unit, DurationUnit.MILLISECONDS)^n        return
durationOfMillis(millis.coerceIn(-MAX_MILLIS, MAX_MILLIS))^n    }^n }^n/^n/** Returns a [Duration] equal
to this [Double] number of the specified [unit].^n *^n * Depending on its magnitude, the value is rounded to an
integer number of nanoseconds or milliseconds.^n *^n * @throws IllegalArgumentException if this `Double` value is
`NaN`.^n *^@SinceKotlin("1.6")^@WasExperimental(ExperimentalTime::class)^npublic fun
Double.toDuration(unit: DurationUnit): Duration {^n    val valueInNs = convertDurationUnit(this, unit,
DurationUnit.NANOSECONDS)^n    require(!valueInNs.isNaN()) { "Duration value cannot be NaN." }^n    val
nanos = valueInNs.roundToLong()^n    return if (nanos in -MAX_NANOS..MAX_NANOS) {^n
durationOfNanos(nanos)^n    } else {^n        val millis = convertDurationUnit(this, unit,
DurationUnit.MILLISECONDS).roundToLong()^n        durationOfMillisNormalized(millis)^n    }^n }^n/^n//
constructing from number of units^// deprecated extension properties^// Returns a [Duration] equal to this [Int]
number of nanoseconds. *^@SinceKotlin("1.3")^@ExperimentalTime^@Deprecated("Use 'Int.nanoseconds'
extension property from Duration.Companion instead.", ReplaceWith("this.nanoseconds",
"\"kotlin.time.Duration.Companion.nanoseconds\""))^@DeprecatedSinceKotlin(warningSince = "1.5")^npublic val
Int.nanoseconds get() = toDuration(DurationUnit.NANOSECONDS)^n/^n/** Returns a [Duration] equal to this
[Long] number of nanoseconds. *^@SinceKotlin("1.3")^@ExperimentalTime^@Deprecated("Use
'Long.nanoseconds' extension property from Duration.Companion instead.", ReplaceWith("this.nanoseconds",
"\"kotlin.time.Duration.Companion.nanoseconds\""))^@DeprecatedSinceKotlin(warningSince = "1.5")^npublic val
Long.nanoseconds get() = toDuration(DurationUnit.NANOSECONDS)^n/^n/** Returns a [Duration] equal to this
[Double] number of nanoseconds.^n *^n * @throws IllegalArgumentException if this [Double] value is `NaN`.^n
*^@SinceKotlin("1.3")^@ExperimentalTime^@Deprecated("Use 'Double.nanoseconds' extension property
from Duration.Companion instead.", ReplaceWith("this.nanoseconds",
"\"kotlin.time.Duration.Companion.nanoseconds\""))^@DeprecatedSinceKotlin(warningSince = "1.5")^npublic val
Double.nanoseconds get() = toDuration(DurationUnit.NANOSECONDS)^n/^n/** Returns a [Duration] equal to
this [Int] number of microseconds. *^@SinceKotlin("1.3")^@ExperimentalTime^@Deprecated("Use
'Int.microseconds' extension property from Duration.Companion instead.", ReplaceWith("this.microseconds",
"\"kotlin.time.Duration.Companion.microseconds\""))^@DeprecatedSinceKotlin(warningSince = "1.5")^npublic val
Int.microseconds get() = toDuration(DurationUnit.MICROSECONDS)^n/^n/** Returns a [Duration] equal to this
[Long] number of microseconds. *^@SinceKotlin("1.3")^@ExperimentalTime^@Deprecated("Use
'Long.microseconds' extension property from Duration.Companion instead.", ReplaceWith("this.microseconds",
"\"kotlin.time.Duration.Companion.microseconds\""))^@DeprecatedSinceKotlin(warningSince = "1.5")^npublic val
Long.microseconds get() = toDuration(DurationUnit.MICROSECONDS)^n/^n/** Returns a [Duration] equal to
this [Double] number of microseconds.^n *^n * @throws IllegalArgumentException if this [Double] value is
`NaN`.^n *^@SinceKotlin("1.3")^@ExperimentalTime^@Deprecated("Use 'Double.microseconds' extension
property from Duration.Companion instead.", ReplaceWith("this.microseconds",
"\"kotlin.time.Duration.Companion.microseconds\""))^@DeprecatedSinceKotlin(warningSince = "1.5")^npublic val
Double.microseconds get() = toDuration(DurationUnit.MICROSECONDS)^n/^n/** Returns a [Duration] equal to
this [Int] number of milliseconds. *^@SinceKotlin("1.3")^@ExperimentalTime^@Deprecated("Use
'Int.milliseconds' extension property from Duration.Companion instead.", ReplaceWith("this.milliseconds",
"\"kotlin.time.Duration.Companion.milliseconds\""))^@DeprecatedSinceKotlin(warningSince = "1.5")^npublic val

```

```

Int.milliseconds get() = toDuration(DurationUnit.MILLISECONDS)\n\n/** Returns a [Duration] equal to this
[Long] number of milliseconds. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use
'Long.milliseconds' extension property from Duration.Companion instead.", ReplaceWith("this.milliseconds",
"\"kotlin.time.Duration.Companion.milliseconds\""))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Long.milliseconds get() = toDuration(DurationUnit.MILLISECONDS)\n\n/**\n * Returns a [Duration] equal to this
[Double] number of milliseconds.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n
*\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Double.milliseconds' extension property
from Duration.Companion instead.", ReplaceWith("this.milliseconds",
"\"kotlin.time.Duration.Companion.milliseconds\""))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Double.milliseconds get() = toDuration(DurationUnit.MILLISECONDS)\n\n/** Returns a [Duration] equal to this
[Int] number of seconds. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Int.seconds'
extension property from Duration.Companion instead.", ReplaceWith("this.seconds",
"\"kotlin.time.Duration.Companion.seconds\""))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Int.seconds get() = toDuration(DurationUnit.SECONDS)\n\n/** Returns a [Duration] equal to this [Long] number of
seconds. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Long.seconds' extension property
from Duration.Companion instead.", ReplaceWith("this.seconds",
"\"kotlin.time.Duration.Companion.seconds\""))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Long.seconds get() = toDuration(DurationUnit.SECONDS)\n\n/**\n * Returns a [Duration] equal to this [Double]
number of seconds.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n
*\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Double.seconds' extension property from
Duration.Companion instead.", ReplaceWith("this.seconds",
"\"kotlin.time.Duration.Companion.seconds\""))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Double.seconds get() = toDuration(DurationUnit.SECONDS)\n\n/** Returns a [Duration] equal to this [Int]
number of minutes. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Int.minutes' extension
property from Duration.Companion instead.", ReplaceWith("this.minutes",
"\"kotlin.time.Duration.Companion.minutes\""))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Int.minutes get() = toDuration(DurationUnit.MINUTES)\n\n/** Returns a [Duration] equal to this [Long] number of
minutes. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Long.minutes' extension property
from Duration.Companion instead.", ReplaceWith("this.minutes",
"\"kotlin.time.Duration.Companion.minutes\""))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Long.minutes get() = toDuration(DurationUnit.MINUTES)\n\n/**\n * Returns a [Duration] equal to this [Double]
number of minutes.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n
*\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Double.minutes' extension property from
Duration.Companion instead.", ReplaceWith("this.minutes",
"\"kotlin.time.Duration.Companion.minutes\""))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Double.minutes get() = toDuration(DurationUnit.MINUTES)\n\n/** Returns a [Duration] equal to this [Int]
number of hours. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Int.hours' extension
property from Duration.Companion instead.", ReplaceWith("this.hours",
"\"kotlin.time.Duration.Companion.hours\""))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Int.hours get() = toDuration(DurationUnit.HOURS)\n\n/** Returns a [Duration] equal to this [Long] number of
hours. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Long.hours' extension property from
Duration.Companion instead.", ReplaceWith("this.hours",
"\"kotlin.time.Duration.Companion.hours\""))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Long.hours get() = toDuration(DurationUnit.HOURS)\n\n/**\n * Returns a [Duration] equal to this [Double]
number of hours.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n
*\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Double.hours' extension property from
Duration.Companion instead.", ReplaceWith("this.hours",
"\"kotlin.time.Duration.Companion.hours\""))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val

```

```

Double.hours get() = toDuration(DurationUnit.HOURS)\n\n/** Returns a [Duration] equal to this [Int] number of
days. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Int.days' extension property from
Duration.Companion instead.", ReplaceWith("this.days"),
"kotlin.time.Duration.Companion.days")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val Int.days
get() = toDuration(DurationUnit.DAYS)\n\n/** Returns a [Duration] equal to this [Long] number of days.
*\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Long.days' extension property from
Duration.Companion instead.", ReplaceWith("this.days"),
"kotlin.time.Duration.Companion.days")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Long.days get() = toDuration(DurationUnit.DAYS)\n\n/** Returns a [Duration] equal to this [Double] number
of days.\n * \n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n
*\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Double.days' extension property from
Duration.Companion instead.", ReplaceWith("this.days"),
"kotlin.time.Duration.Companion.days")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Double.days get() = toDuration(DurationUnit.DAYS)\n\n/** Returns a duration whose value is the specified
[duration] value multiplied by this number.
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalTime::class)\n@kotlin.internal.InlineOnly\npublic
inline operator fun Int.times(duration: Duration): Duration = duration * this\n\n/** Returns a duration whose
value is the specified [duration] value multiplied by this number.\n * \n * The operation may involve rounding when
the result cannot be represented exactly with a [Double] number.\n * \n * @throws IllegalArgumentException if the
operation results in a `NaN` value.\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalTime::class)\n@kotlin.internal.InlineOnly\npublic
inline operator fun Double.times(duration: Duration): Duration = duration * this\n\n\nprivate fun
parseDuration(value: String, strictIso: Boolean): Duration {\n    var length = value.length\n    if (length == 0) throw
IllegalArgumentException("The string is empty")\n    var index = 0\n    var result = Duration.ZERO\n    val
infinityString = "Infinity"\n    when (value[index]) {\n        '+', '-' -> index++\n    }\n    val hasSign = index > 0\n
val isNegative = hasSign && value.startsWith('-')\n    when {\n        length <= index ->\n            throw
IllegalArgumentException("No components")\n        value[index] == 'P' -> {\n            if (++index == length) throw
IllegalArgumentException()\n            val nonDigitSymbols = "+-." \n            var isTimeComponent = false\n
var prevUnit: DurationUnit? = null\n            while (index < length) {\n                if (value[index] == 'T') {\n
                    if (isTimeComponent || ++index == length) throw IllegalArgumentException()\n                    isTimeComponent =
true\n                    continue\n                }\n                val component = value.substringWhile(index) { it in '0'..'9' || it in
nonDigitSymbols }\n                if (component.isEmpty()) throw IllegalArgumentException()\n                index +=
component.length\n                val unitChar = value.getOrElse(index) { throw IllegalArgumentException("Missing
unit for value $component") }\n                index++\n                val unit = durationUnitByIsoChar(unitChar,
isTimeComponent)\n                if (prevUnit != null && prevUnit <= unit) throw
IllegalArgumentException("Unexpected order of duration components")\n                prevUnit = unit\n                val
dotIndex = component.indexOf('.')\n                if (unit == DurationUnit.SECONDS && dotIndex > 0) {\n
                    val whole = component.substring(0, dotIndex)\n                    result +=
parseOverLongIsoComponent(whole).toDuration(unit)\n                    result +=
component.substring(dotIndex).toDouble().toDuration(unit)\n                } else {\n                    result +=
parseOverLongIsoComponent(component).toDuration(unit)\n                }\n            }\n            strictIso ->\n                throw
IllegalArgumentException("value.regionMatches(index, infinityString, 0, length = maxOf(length -
index, infinityString.length), ignoreCase = true) -> {\n                    result = Duration.INFINITE\n                }\n            else -> {\n
                // parse default string format\n                var prevUnit: DurationUnit? = null\n                var afterFirst = false\n
var allowSpaces = !hasSign\n                if (hasSign && value[index] == '(' && value.last() == ')') {\n
                    allowSpaces = true\n                    if (++index == --length) throw
IllegalArgumentException("No components")\n                }\n                while (index < length) {\n                    if (afterFirst && allowSpaces) {\n                        index =
value.skipWhile(index) { it == ' ' }\n                    }\n                    afterFirst = true\n                    val component =

```



```

ULong(0) else ULong(1)\n } \n // Optimization - use signed division if both dividend and divisor < 2^63\n if
(dividend >= 0) {\n return ULong(dividend / divisor)\n } \n // Otherwise, approximate the quotient, check,
and correct if necessary.\n val quotient = ((dividend ushr 1) / divisor) shl 1\n val rem = dividend - quotient *
divisor\n return ULong(quotient + if (ULong(rem) >= ULong(divisor)) 1 else 0)\n}\n\n@PublishedApi\ninternal
fun ulongRemainder(v1: ULong, v2: ULong): ULong {\n val dividend = v1.toLong()\n val divisor =
v2.toLong()\n if (divisor < 0) { // i.e., divisor >= 2^63:\n return if (v1 < v2) {\n v1 // dividend <
divisor\n } else {\n v1 - v2 // dividend >= divisor\n }\n } \n // Optimization - use signed
modulus if both dividend and divisor < 2^63\n if (dividend >= 0) {\n return ULong(dividend % divisor)\n
}\n // Otherwise, approximate the quotient, check, and correct if necessary.\n val quotient = ((dividend ushr 1)
/ divisor) shl 1\n val rem = dividend - quotient * divisor\n return ULong(rem - if (ULong(rem) >=
ULong(divisor)) divisor else 0)\n}\n\n@PublishedApi\ninternal fun doubleToUInt(v: Double): UInt = when {\n
v.isNaN() -> 0u\n v <= UInt.MIN_VALUE.toDouble() -> UInt.MIN_VALUE\n v >=
UInt.MAX_VALUE.toDouble() -> UInt.MAX_VALUE\n v <= Int.MAX_VALUE -> v.toInt().toUInt()\n else -
> (v - Int.MAX_VALUE).toInt().toUInt() + Int.MAX_VALUE.toUInt() // Int.MAX_VALUE < v <
UInt.MAX_VALUE\n}\n\n@PublishedApi\ninternal fun doubleToULong(v: Double): ULong = when {\n
v.isNaN() -> 0u\n v <= ULong.MIN_VALUE.toDouble() -> ULong.MIN_VALUE\n v >=
ULong.MAX_VALUE.toDouble() -> ULong.MAX_VALUE\n v < Long.MAX_VALUE ->
v.toLong().toULong()\n // Real values from Long.MAX_VALUE to (Long.MAX_VALUE + 1) are not
representable in Double, so don't handle them.\n else -> (v - 9223372036854775808.0).toLong().toULong() +
9223372036854775808uL // Long.MAX_VALUE + 1 < v <
ULong.MAX_VALUE\n}\n\n\n@PublishedApi\ninternal fun uintToDouble(v: Int): Double = (v and
Int.MAX_VALUE).toDouble() + (v ushr 31 shl 30).toDouble() * 2\n\n@PublishedApi\ninternal fun
ulongToDouble(v: Long): Double = (v ushr 11).toDouble() * 2048 + (v and 2047)\n\n\ninternal fun
ulongToString(v: Long): String = ulongToString(v, 10)\n\n\ninternal fun ulongToString(v: Long, base: Int): String {\n
if (v >= 0) return v.toString(base)\n var quotient = ((v ushr 1) / base) shl 1\n var rem = v - quotient * base\n
if (rem >= base) {\n rem -= base\n quotient += 1\n }\n return quotient.toString(base) +
rem.toString(base)\n}\n\n", /*\n * Copyright 2010-2016 JetBrains s.r.o.\n * Licensed under the Apache License,
Version 2.0 (the "License");\n * you may not use this file except in compliance with the License.\n * You may
obtain a copy of the License at\n * http://www.apache.org/licenses/LICENSE-2.0\n * Unless required by
applicable law or agreed to in writing, software\n * distributed under the License is distributed on an "AS IS"
BASIS,\n * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.\n * See the
License for the specific language governing permissions and\n * limitations under the License.\n */\n\npackage
kotlin.internal\n\n/**\n * Specifies that the corresponding type parameter is not used for unsafe operations such as
casts or 'is' checks\n * That means it's completely safe to use generic types as argument for such parameter.\n
*/\n\n@Target(AnnotationTarget.TYPE_PARAMETER)\n@Retention(AnnotationRetention.BINARY)\ninternal
annotation class PureReifiable\n\n/**\n * Specifies that the corresponding built-in method exists depending on
platform.\n * Current implementation for JVM looks whether method with same JVM descriptor exists in the
module JDK.\n * For example MutableMap.remove(K, V) available only if corresponding\n * method
'java/util/Map.remove(Ljava/lang/Object;Ljava/lang/Object;)Z' is defined in JDK (i.e. for major versions >= 8)\n
*/\n\n@Target(AnnotationTarget.FUNCTION)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation
class PlatformDependent\n\n/**\n * When applied to a function or property, enables a compiler optimization that
evaluates that function or property\n * at compile-time and replaces calls to it with the computed result.\n
*/\n\n@Target(AnnotationTarget.CONSTRUCTOR, AnnotationTarget.FUNCTION,
AnnotationTarget.PROPERTY)\n@Retention(AnnotationRetention.BINARY)\n@SinceKotlin("1.7")\ninternal
annotation class IntrinsicConstEvaluation\n", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("CollectionsKt")\n\npackage

```

```

kotlin.collections\n\n/**\n * Given an [iterator] function constructs an [Iterable] instance that returns values through
the [Iterator]\n * provided by that function.\n * @sample samples.collections.Iterables.Building.iterable\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable(crossinline iterator: () -> Iterator<T>): Iterable<T>
= object : Iterable<T> {\n    override fun iterator(): Iterator<T> = iterator()\n}\n\n/**\n * A wrapper over another
[Iterable] (or any other object that can produce an [Iterator]) that returns\n * an indexing iterator.\n */\ninternal class
IndexingIterable<out T>(private val iteratorFactory: () -> Iterator<T>) : Iterable<IndexedValue<T>> {\n    override
fun iterator(): Iterator<IndexedValue<T>> = IndexingIterator(iteratorFactory())\n}\n\n/**\n * Returns the size of
this iterable if it is known, or `null` otherwise.\n */\n@PublishedApi\ninternal fun <T>
Iterable<T>.collectionSizeOrNull(): Int? = if (this is Collection<*>) this.size else null\n\n/**\n * Returns the size of
this iterable if it is known, or the specified [default] value otherwise.\n */\n@PublishedApi\ninternal fun <T>
Iterable<T>.collectionSizeOrDefault(default: Int): Int = if (this is Collection<*>) this.size else default\n\n/**\n *
Returns a single list of all elements from all collections in the given collection.\n * @sample
samples.collections.Iterables.Operations.flattenIterable\n */\npublic fun <T> Iterable<Iterable<T>>.flatten():
List<T> {\n    val result = ArrayList<T>()\n    for (element in this) {\n        result.addAll(element)\n    }\n    return
result\n}\n\n/**\n * Returns a pair of lists, where\n * *first* list is built from the first values of each pair from this
collection,\n * *second* list is built from the second values of each pair from this collection.\n * @sample
samples.collections.Iterables.Operations.unzipIterable\n */\npublic fun <T, R> Iterable<Pair<T, R>>.unzip():
Pair<List<T>, List<R>> {\n    val expectedSize = collectionSizeOrDefault(10)\n    val listT =
ArrayList<T>(expectedSize)\n    val listR = ArrayList<R>(expectedSize)\n    for (pair in this) {\n
listT.add(pair.first)\n        listR.add(pair.second)\n    }\n    return listT to listR\n}\n\n"/*\n * Copyright 2010-2020
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("SequencesKt")\n\npackage
kotlin.sequences\n\nimport kotlin.random.Random\n\n/**\n * Given an [iterator] function constructs a [Sequence]
that returns values through the [Iterator]\n * provided by that function.\n * The values are evaluated lazily, and the
sequence is potentially infinite.\n */\n * @sample samples.collections.Sequences.Building.sequenceFromIterator\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence(crossinline iterator: () -> Iterator<T>):
Sequence<T> = object : Sequence<T> {\n    override fun iterator(): Iterator<T> = iterator()\n}\n\n/**\n * Creates a
sequence that returns all elements from this iterator. The sequence is constrained to be iterated only once.\n */\n *
@sample samples.collections.Sequences.Building.sequenceFromIterator\n */\npublic fun <T>
Iterator<T>.asSequence(): Sequence<T> = Sequence { this }.constrainOnce()\n\n/**\n * Creates a sequence that
returns the specified values.\n */\n * @sample samples.collections.Sequences.Building.sequenceOfValues\n
*/\npublic fun <T> sequenceOf(vararg elements: T): Sequence<T> = if (elements.isEmpty()) emptySequence() else
elements.asSequence()\n\n/**\n * Returns an empty sequence.\n */\npublic fun <T> emptySequence():
Sequence<T> = EmptySequence\n\nprivate object EmptySequence : Sequence<Nothing>,\n
DropTakeSequence<Nothing> {\n    override fun iterator(): Iterator<Nothing> = EmptyIterator\n    override fun
drop(n: Int) = EmptySequence\n    override fun take(n: Int) = EmptySequence\n}\n\n/**\n * Returns this sequence if
it's not `null` and the empty sequence otherwise.\n */\n * @sample
samples.collections.Sequences.Usage.sequenceOrEmpty\n
*/\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>?.orEmpty():
Sequence<T> = this ?: emptySequence()\n\n/**\n * Returns a sequence that iterates through the elements either of
this sequence\n * or, if this sequence turns out to be empty, of the sequence returned by [defaultValue] function.\n
*/\n * @sample samples.collections.Sequences.Usage.sequenceIfEmpty\n */\n@SinceKotlin("1.3")\npublic fun
<T> Sequence<T>.ifEmpty(defaultValue: () -> Sequence<T>): Sequence<T> = sequence {\n    val iterator =
this@ifEmpty.iterator()\n    if (iterator.hasNext()) {\n        yieldAll(iterator)\n    } else {\n
yieldAll(defaultValue())\n    }\n}\n\n/**\n * Returns a sequence of all elements from all sequences in this
sequence.\n */\n * The operation is _intermediate_ and _stateless_.\n */\n * @sample
samples.collections.Sequences.Transformations.flattenSequenceOfSequences\n */\npublic fun <T>

```



```

[IndexValue] objects.\n *\ninternal class IndexingSequence<T>\nconstructor(private val sequence:
Sequence<T>) : Sequence<IndexedValue<T>> {\n  override fun iterator(): Iterator<IndexedValue<T>> = object :
Iterator<IndexedValue<T>> {\n    val iterator = sequence.iterator()\n    var index = 0\n    override fun next():
IndexedValue<T> {\n      return IndexedValue(checkIndexOverflow(index++), iterator.next())\n    }\n\n  override fun hasNext(): Boolean {\n    return iterator.hasNext()\n  }\n}\n}\n\n**\n * A sequence which
takes the values from two parallel underlying sequences, passes them to the given\n * [transform] function and
returns the values returned by that function. The sequence stops returning\n * values as soon as one of the
underlying sequences stops returning values.\n *\ninternal class MergingSequence<T1, T2, V>\nconstructor(\nprivate val sequence1: Sequence<T1>,\nprivate val sequence2: Sequence<T2>,\nprivate val transform: (T1,
T2) -> V)\n : Sequence<V> {\n  override fun iterator(): Iterator<V> = object : Iterator<V> {\n    val iterator1 =
sequence1.iterator()\n    val iterator2 = sequence2.iterator()\n    override fun next(): V {\n      return
transform(iterator1.next(), iterator2.next())\n    }\n\n    override fun hasNext(): Boolean {\n      return
iterator1.hasNext() && iterator2.hasNext()\n    }\n  }\n}\n\ninternal class FlatteningSequence<T, R,
E>\nconstructor(\nprivate val sequence: Sequence<T>,\nprivate val transformer: (T) -> R,\nprivate val
iterator: (R) -> Iterator<E>)\n : Sequence<E> {\n  override fun iterator(): Iterator<E> = object : Iterator<E> {\n
val iterator = sequence.iterator()\n    var itemIterator: Iterator<E>? = null\n\n    override fun next(): E {\n
if (!ensureItemIterator())\n      throw NoSuchElementException()\n      return itemIterator!!.next()\n
}\n\n    override fun hasNext(): Boolean {\n      return ensureItemIterator()\n    }\n\n    private fun
ensureItemIterator(): Boolean {\n      if (itemIterator?.hasNext() == false)\n        itemIterator = null\n\n      while (itemIterator == null) {\n        if (!iterator.hasNext()) {\n          return false\n        } else {\n
val element = iterator.next()\n          val nextItemIterator = iterator(transformer(element))\n\n          if (nextItemIterator.hasNext()) {\n            itemIterator = nextItemIterator\n            return true\n          }\n        }\n      }\n      return true\n    }\n  }\n}\n\ninternal fun <T, C, R> flatMapIndexed(source:
Sequence<T>, transform: (Int, T) -> C, iterator: (C) -> Iterator<R>): Sequence<R> =\nsequence {\n  var
index = 0\n  for (element in source) {\n    val result = transform(checkIndexOverflow(index++), element)\n
yieldAll(iterator(result))\n  }\n}\n\n**\n * A sequence that supports drop(n) and take(n) operations\n
*\ninternal interface DropTakeSequence<T> : Sequence<T> {\n  fun drop(n: Int): Sequence<T>\n  fun take(n:
Int): Sequence<T>\n}\n\n**\n * A sequence that skips [startIndex] values from the underlying [sequence]\n * and
stops returning values right before [endIndex], i.e. stops at `endIndex - 1`\n *\ninternal class SubSequence<T>(\nprivate val sequence: Sequence<T>,\nprivate val startIndex: Int,\nprivate val endIndex: Int)\n : Sequence<T>,\nDropTakeSequence<T> {\n\n  init {\n    require(startIndex >= 0) { "\"startIndex should be non-negative, but is
$startIndex\""}\n    require(endIndex >= 0) { "\"endIndex should be non-negative, but is $endIndex\""}\n
require(endIndex >= startIndex) { "\"endIndex should be not less than startIndex, but was $endIndex < $startIndex\""}\n
}\n\n  private val count: Int get() = endIndex - startIndex\n\n  override fun drop(n: Int): Sequence<T> = if (n
>= count) emptySequence() else SubSequence(sequence, startIndex + n, endIndex)\n\n  override fun take(n: Int):
Sequence<T> = if (n >= count) this else SubSequence(sequence, startIndex, startIndex + n)\n\n  override fun
iterator() = object : Iterator<T> {\n\n    val iterator = sequence.iterator()\n    var position = 0\n\n    //
Shouldn't be called from constructor to avoid premature iteration\n    private fun drop() {\n      while (position
< startIndex && iterator.hasNext()) {\n        iterator.next()\n        position++\n      }\n    }\n\n    override fun hasNext(): Boolean {\n      drop()\n      return (position < endIndex) && iterator.hasNext()\n
}\n\n    override fun next(): T {\n      drop()\n      if (position >= endIndex)\n        throw
NoSuchElementException()\n      position++\n      return iterator.next()\n    }\n  }\n}\n\n**\n * A
sequence that returns at most [count] values from the underlying [sequence], and stops returning values\n * as soon
as that count is reached.\n *\ninternal class TakeSequence<T>(\nprivate val sequence: Sequence<T>,\nprivate val
count: Int)\n : Sequence<T>,\nDropTakeSequence<T> {\n\n  init {\n    require(count >= 0) { "\"count must be
non-negative, but was $count.\""}\n  }\n\n  override fun drop(n: Int): Sequence<T> = if (n >= count)
emptySequence() else SubSequence(sequence, n, count)\n\n  override fun take(n: Int): Sequence<T> = if (n >=
count) this else TakeSequence(sequence, n)\n\n  override fun iterator(): Iterator<T> = object : Iterator<T> {\n

```

```

var left = count\n    val iterator = sequence.iterator()\n\n    override fun next(): T {\n        if (left == 0)\n            throw NoSuchElementException()\n        left--\n        return iterator.next()\n    }\n\n    override fun\n    hasNext(): Boolean {\n        return left > 0 && iterator.hasNext()\n    }\n}\n\n/**\n * A sequence that\n    returns values from the underlying [sequence] while the [predicate] function returns\n    * `true`, and stops returning\n    values once the function returns `false` for the next element.\n */\n\ninternal class\n    TakeWhileSequence<T>\n        constructor(\n            private val sequence: Sequence<T>,\n            private val predicate: (T) ->\n                Boolean\n        ) : Sequence<T> {\n            override fun iterator(): Iterator<T> = object : Iterator<T> {\n                val iterator =\n                    sequence.iterator()\n                var nextState: Int = -1 // -1 for unknown, 0 for done, 1 for continue\n                var nextItem: T?\n                    = null\n\n                private fun calcNext() {\n                    if (iterator.hasNext()) {\n                        val item = iterator.next()\n                        if (predicate(item)) {\n                            nextState = 1\n                            nextItem = item\n                            return\n                        }\n                        nextState = 0\n                    }\n\n                    override fun next(): T {\n                        if (nextState == -1)\n                            calcNext() // will change nextState\n                        if (nextState == 0)\n                            throw NoSuchElementException()\n                        @Suppress("UNCHECKED_CAST")\n                            val result = nextItem as T\n                            // Clean next to avoid keeping\n                            reference on yielded instance\n                            nextItem = null\n                            nextState = -1\n                            return result\n                        }\n\n                    override fun hasNext(): Boolean {\n                        if (nextState == -1)\n                            calcNext() // will change nextState\n                        return nextState == 1\n                    }\n                }\n\n            }\n\n            /**\n             * A sequence that skips the specified number of values from the\n                underlying [sequence] and returns\n                * all values after that.\n            */\n\n            internal class DropSequence<T>(\n                private val\n                    sequence: Sequence<T>,\n                private val count: Int\n            ) : Sequence<T>, DropTakeSequence<T> {\n                init {\n                    require(count >= 0) { "\n                    count must be non-negative, but was $count.\n                }\n                }\n\n                override fun drop(n: Int):\n                    Sequence<T> = (count + n).let { n1 -> if (n1 < 0) DropSequence(this, n)\n                    else DropSequence(sequence, n1) }\n\n                override fun take(n: Int): Sequence<T> = (count + n).let { n1 -> if (n1 < 0) TakeSequence(this, n)\n                else\n                    SubSequence(sequence, count, n1) }\n\n                override fun iterator(): Iterator<T> = object : Iterator<T> {\n                    val\n                        iterator = sequence.iterator()\n                    var left = count\n                    // Shouldn't be called from constructor to avoid\n                    premature iteration\n                    private fun drop() {\n                        while (left > 0 && iterator.hasNext()) {\n                            iterator.next()\n                            left--\n                        }\n                    }\n\n                    override fun next(): T {\n                        drop()\n                        return\n                            iterator.next()\n                    }\n\n                    override fun hasNext(): Boolean {\n                        drop()\n                        return iterator.hasNext()\n                    }\n                }\n\n            }\n\n            /**\n             * A sequence that skips the values from the underlying [sequence] while the given\n                [predicate] returns `true` and returns\n                * all values after that.\n            */\n\n            internal class\n                DropWhileSequence<T>\n                    constructor(\n                        private val sequence: Sequence<T>,\n                        private val predicate: (T) ->\n                            Boolean\n                    ) : Sequence<T> {\n                        override fun iterator(): Iterator<T> = object : Iterator<T> {\n                            val iterator =\n                                sequence.iterator()\n                            var dropState: Int = -1 // -1 for not dropping, 1 for nextItem, 0 for normal iteration\n                            var nextItem: T?\n                                = null\n\n                            private fun drop() {\n                                while (iterator.hasNext()) {\n                                    val item =\n                                        iterator.next()\n                                    if (!predicate(item)) {\n                                        nextItem = item\n                                        dropState = 1\n                                        return\n                                    }\n                                }\n                                dropState = 0\n                            }\n\n                            override fun next(): T {\n                                if\n                                    (dropState == -1)\n                                    drop()\n                                if (dropState == 1) {\n                                    @Suppress("UNCHECKED_CAST")\n                                        val result = nextItem as T\n                                        nextItem = null\n                                        dropState = 0\n                                        return result\n                                    }\n                                return iterator.next()\n                            }\n\n                            override fun\n                                hasNext(): Boolean {\n                                    if (dropState == -1)\n                                        drop()\n                                    return dropState == 1 ||\n                                        iterator.hasNext()\n                                }\n                        }\n\n                    internal class DistinctSequence<T, K>(\n                        private val source: Sequence<T>,\n                        private val keySelector: (T) -> K\n                    ) : Sequence<T> {\n                        override fun iterator(): Iterator<T> =\n                            DistinctIterator(source.iterator(), keySelector)\n                    }\n\n                    private class DistinctIterator<T, K>(\n                        private val source:\n                            Iterator<T>,\n                        private val keySelector: (T) -> K\n                    ) : AbstractIterator<T>() {\n                        private val observed =\n                            HashSet<K>()\n                        override fun computeNext() {\n                            while (source.hasNext()) {\n                                val next =\n                                    source.next()\n                                val key = keySelector(next)\n                                if (observed.add(key)) {\n                                    setNext(next)\n                                    return\n                                }\n                            }\n                            done()\n                        }\n                    }\n\n                    private class GeneratorSequence<T : Any>(\n                        private val\n                            getInitialValue: () -> T?,\n                        private val getNextValue: (T) -> T? : Sequence<T> {\n                            override fun iterator():\n                                Iterator<T> = object : Iterator<T> {\n                                    var nextItem: T?\n                                        = null\n                                    var nextState: Int = -2 // -2 for initial\n                                    unknown, -1 for next unknown, 0 for done, 1 for continue\n                                    private fun calcNext() {\n                                        nextItem = if

```

```

(nextState == -2) getInitialValue() else getNextValue(nextItem!!)\n        nextState = if (nextItem == null) 0 else
1\n    }\n\n    override fun next(): T {\n        if (nextState < 0)\n            calcNext()\n        if (nextState
== 0)\n            throw NoSuchElementException()\n        val result = nextItem as T\n        // Do not clean
nextItem (to avoid keeping reference on yielded instance) -- need to keep state for getNextValue\n        nextState
= -1\n        return result\n    }\n\n    override fun hasNext(): Boolean {\n        if (nextState < 0)\n            calcNext()\n        return nextState == 1\n    }\n}\n\n/**\n * Returns a wrapper sequence that provides
values of this sequence, but ensures it can be iterated only one time.\n *\n * The operation is _intermediate_ and
_stateless_.  

 * [IllegalStateException] is thrown on iterating the returned sequence for the second time and the
following times.\n *\n */\npublic fun <T> Sequence<T>.constrainOnce(): Sequence<T> {\n    // as? does not work
in js\n    //return this as? ConstrainedOnceSequence<T> ?: ConstrainedOnceSequence(this)\n    return if (this is
ConstrainedOnceSequence<T>) this else ConstrainedOnceSequence(this)\n}\n\n/**\n * Returns a sequence which
invokes the function to calculate the next value on each iteration until the function returns `null`.\n *\n * The
returned sequence is constrained to be iterated only once.\n *\n * @see constrainOnce\n *\n */\npublic fun
kotlin.sequences.sequence\n *\n * @sample samples.collections.Sequences.Building.generateSequence\n */\npublic fun
<T : Any> generateSequence(nextFunction: () -> T?): Sequence<T> {\n    return
GeneratorSequence(nextFunction, { nextFunction() }).constrainOnce()\n}\n\n/**\n * Returns a sequence defined by
the starting value [seed] and the function [nextFunction],\n *\n * which is invoked to calculate the next value based on
the previous one on each iteration.\n *\n * The sequence produces values until it encounters first `null` value.\n *\n * If
[seed] is `null`, an empty sequence is produced.\n *\n * The sequence can be iterated multiple times, each time
starting with [seed].\n *\n * @see kotlin.sequences.sequence\n *\n * @sample
samples.collections.Sequences.Building.generateSequenceWithSeed\n */\n@kotlin.internal.LowPriorityInOverloadResolution\npublic fun <T : Any> generateSequence(seed: T?,
nextFunction: (T) -> T?): Sequence<T> =\n    if (seed == null)\n        EmptySequence\n    else\n        GeneratorSequence({ seed }, nextFunction)\n}\n\n/**\n * Returns a sequence defined by the function [seedFunction],
which is invoked to produce the starting value,\n *\n * and the [nextFunction], which is invoked to calculate the next
value based on the previous one on each iteration.\n *\n * The sequence produces values until it encounters first
`null` value.\n *\n * If [seedFunction] returns `null`, an empty sequence is produced.\n *\n * The sequence can be
iterated multiple times.\n *\n * @see kotlin.sequences.sequence\n *\n * @sample
samples.collections.Sequences.Building.generateSequenceWithLazySeed\n */\npublic fun <T : Any>
generateSequence(seedFunction: () -> T?, nextFunction: (T) -> T?): Sequence<T> =\n    GeneratorSequence(seedFunction, nextFunction)\n}\n\n"/\n *\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n *\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("PreconditionsKt")\n\npackage
kotlin\n\nimport kotlin.contracts.contract\n\n/**\n * Throws an [IllegalArgumentException] if the [value] is false.\n *\n * @sample samples.misc.Preconditions.failRequireWithLazyMessage\n */\n@kotlin.internal.InlineOnly\npublic
inline fun require(value: Boolean): Unit {\n    contract {\n        returns() implies value\n    }\n    require(value) {\n
"Failed requirement." }\n}\n\n/**\n * Throws an [IllegalArgumentException] with the result of calling
[lazyMessage] if the [value] is false.\n *\n * @sample samples.misc.Preconditions.failRequireWithLazyMessage\n */\n@kotlin.internal.InlineOnly\npublic inline fun require(value: Boolean, lazyMessage: () -> Any): Unit {\n
contract {\n        returns() implies value\n    }\n    if (!value) {\n        val message = lazyMessage()\n        throw
IllegalArgumentException(message.toString())\n    }\n}\n\n/**\n * Throws an [IllegalArgumentException] if the
[value] is null. Otherwise returns the not null value.\n *\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T : Any>
requireNotNull(value: T?): T {\n    contract {\n        returns() implies (value != null)\n    }\n    return
requireNotNull(value) {"Required value was null." }\n}\n\n/**\n * Throws an [IllegalArgumentException] with
the result of calling [lazyMessage] if the [value] is null. Otherwise\n *\n * returns the not null value.\n *\n * @sample
samples.misc.Preconditions.failRequireNotNullWithLazyMessage\n */\n@kotlin.internal.InlineOnly\npublic inline fun
<T : Any> requireNotNull(value: T?, lazyMessage: () -> Any): T {\n    contract {\n        returns() implies (value

```

```

!= null)\n } \n\n if (value == null) {\n     val message = lazyMessage()\n     throw
IllegalArgumentException(message.toString())\n } else {\n     return value\n }\n}\n\n/**\n * Throws an
[IllegalStateException] if the [value] is false.\n *\n * @sample
samples.misc.Preconditions.failCheckWithLazyMessage\n */\n@kotlin.internal.InlineOnly\npublic inline fun
check(value: Boolean): Unit {\n    contract {\n        returns() implies value\n    }\n    check(value) { \"Check failed.\"
}\n}\n\n/**\n * Throws an [IllegalStateException] with the result of calling [lazyMessage] if the [value] is false.\n
*\n * @sample samples.misc.Preconditions.failCheckWithLazyMessage\n */\n@kotlin.internal.InlineOnly\npublic
inline fun check(value: Boolean, lazyMessage: () -> Any): Unit {\n    contract {\n        returns() implies value\n    }\n
    if (!value) {\n        val message = lazyMessage()\n        throw IllegalStateException(message.toString())\n
    }\n}\n\n/**\n * Throws an [IllegalStateException] if the [value] is null. Otherwise\n * returns the not null value.\n
*\n * @sample samples.misc.Preconditions.failCheckWithLazyMessage\n */\n@kotlin.internal.InlineOnly\npublic
inline fun <T : Any> checkNotNull(value: T?): T {\n    contract {\n        returns() implies (value != null)\n    }\n
    return checkNotNull(value) { \"Required value was null.\" }\n}\n\n/**\n * Throws an [IllegalStateException] with
the result of calling [lazyMessage] if the [value] is null. Otherwise\n * returns the not null value.\n *\n * @sample
samples.misc.Preconditions.failCheckWithLazyMessage\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T :
Any> checkNotNull(value: T?, lazyMessage: () -> Any): T {\n    contract {\n        returns() implies (value != null)\n
    }\n\n    if (value == null) {\n        val message = lazyMessage()\n        throw
IllegalStateException(message.toString())\n    } else {\n        return value\n    }\n}\n\n/**\n * Throws an
[IllegalStateException] with the given [message].\n *\n * @sample samples.misc.Preconditions.failWithError\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun error(message: Any): Nothing = throw
IllegalStateException(message.toString())\n\"\", /*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\n// NOTE: THIS FILE IS AUTO-GENERATED
by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport
kotlin.js.*\nimport primitiveArrayConcat\nimport withType\nimport kotlin.ranges.contains\nimport
kotlin.ranges.reversed\n\n/**\n * Returns an element at the given [index] or throws an
[IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n *\n * @sample
samples.collections.Collections.Elements.elementAt\n */\n\npublic actual fun <T> Array<out T>.elementAt(index:
Int): T {\n    return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, size: $size\") }\n
}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
out of bounds of this array.\n *\n * @sample samples.collections.Collections.Elements.elementAt\n */\n\npublic
actual fun ByteArray.elementAt(index: Int): Byte {\n    return elementAtOrElse(index) { throw
IndexOutOfBoundsException(\"index: $index, size: $size\") }\n}\n\n/**\n * Returns an element at the given
[index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n *\n * @sample
samples.collections.Collections.Elements.elementAt\n */\n\npublic actual fun ShortArray.elementAt(index: Int): Short
{\n    return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, size: $size\") }\n
}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
out of bounds of this array.\n *\n * @sample samples.collections.Collections.Elements.elementAt\n */\n\npublic
actual fun IntArray.elementAt(index: Int): Int {\n    return elementAtOrElse(index) { throw
IndexOutOfBoundsException(\"index: $index, size: $size\") }\n}\n\n/**\n * Returns an element at the given
[index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n *\n * @sample
samples.collections.Collections.Elements.elementAt\n */\n\npublic actual fun LongArray.elementAt(index: Int): Long
{\n    return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, size: $size\") }\n
}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
out of bounds of this array.\n *\n * @sample samples.collections.Collections.Elements.elementAt\n */\n\npublic
actual fun FloatArray.elementAt(index: Int): Float {\n    return elementAtOrElse(index) { throw
IndexOutOfBoundsException(\"index: $index, size: $size\") }\n}\n\n/**\n * Returns an element at the given
[index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n *\n * @sample

```

```

samples.collections.Collections.Elements.elementAt\n *^\\npublic actual fun DoubleArray.elementAt(index: Int):
Double {\\n    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size")}\\n
}\\n}\\n\\n/**\\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
out of bounds of this array.\\n * \\n * @sample samples.collections.Collections.Elements.elementAt\n *^\\npublic
actual fun BooleanArray.elementAt(index: Int): Boolean {\\n    return elementAtOrElse(index) { throw
IndexOutOfBoundsException("index: $index, size: $size")}\\n}\\n}\\n\\n/**\\n * Returns an element at the given
[index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\\n * \\n * @sample
samples.collections.Collections.Elements.elementAt\n *^\\npublic actual fun CharArray.elementAt(index: Int): Char
{\\n    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size")}\\n
}\\n}\\n\\n/**\\n * Returns a [List] that wraps the original array.\\n *^\\npublic actual fun <T> Array<out T>.asList():
List<T> {\\n    return ArrayList<T>(this.unsafeCast<Array<Any?>>())\\n}\\n}\\n\\n/**\\n * Returns a [List] that wraps the
original array.\\n *^\\n@kotlin.internal.InlineOnly\\npublic actual inline fun ByteArray.asList(): List<Byte> {\\n
return this.unsafeCast<Array<Byte>>().asList()\\n}\\n}\\n\\n/**\\n * Returns a [List] that wraps the original array.\\n
*^\\n@kotlin.internal.InlineOnly\\npublic actual inline fun ShortArray.asList(): List<Short> {\\n    return
this.unsafeCast<Array<Short>>().asList()\\n}\\n}\\n\\n/**\\n * Returns a [List] that wraps the original array.\\n
*^\\n@kotlin.internal.InlineOnly\\npublic actual inline fun IntArray.asList(): List<Int> {\\n    return
this.unsafeCast<Array<Int>>().asList()\\n}\\n}\\n\\n/**\\n * Returns a [List] that wraps the original array.\\n
*^\\n@kotlin.internal.InlineOnly\\npublic actual inline fun LongArray.asList(): List<Long> {\\n    return
this.unsafeCast<Array<Long>>().asList()\\n}\\n}\\n\\n/**\\n * Returns a [List] that wraps the original array.\\n
*^\\n@kotlin.internal.InlineOnly\\npublic actual inline fun FloatArray.asList(): List<Float> {\\n    return
this.unsafeCast<Array<Float>>().asList()\\n}\\n}\\n\\n/**\\n * Returns a [List] that wraps the original array.\\n
*^\\n@kotlin.internal.InlineOnly\\npublic actual inline fun DoubleArray.asList(): List<Double> {\\n    return
this.unsafeCast<Array<Double>>().asList()\\n}\\n}\\n\\n/**\\n * Returns a [List] that wraps the original array.\\n
*^\\n@kotlin.internal.InlineOnly\\npublic actual inline fun BooleanArray.asList(): List<Boolean> {\\n    return
this.unsafeCast<Array<Boolean>>().asList()\\n}\\n}\\n\\n/**\\n * Returns a [List] that wraps the original array.\\n
*^\\npublic actual fun CharArray.asList(): List<Char> {\\n    return object : AbstractList<Char>(), RandomAccess {\\n
        override val size: Int get() = this@asList.size\\n        override fun isEmpty(): Boolean = this@asList.isEmpty()\\n
        override fun contains(element: Char): Boolean = this@asList.contains(element)\\n        override fun get(index: Int):
Char {\\n            AbstractList.checkElementIndex(index, size)\\n            return this@asList[index]\\n        }\\n
        override fun indexOf(element: Char): Int {\\n            @Suppress("USELESS_CAST")\\n            if ((element as
Any?) !is Char) return -1\\n            return this@asList.indexOf(element)\\n        }\\n        override fun
lastIndexOf(element: Char): Int {\\n            @Suppress("USELESS_CAST")\\n            if ((element as Any?) !is
Char) return -1\\n            return this@asList.lastIndexOf(element)\\n        }\\n    }\\n}\\n}\\n\\n/**\\n * Returns `true` if the
two specified arrays are *deeply* equal to one another.\\n * i.e. contain the same number of the same elements in the
same order.\\n * \\n * If two corresponding elements are nested arrays, they are also compared deeply.\\n * If any of
arrays contains itself on any nesting level the behavior is undefined.\\n * \\n * The elements of other types are
compared for equality with the [equals][Any.equals] function.\\n * For floating point numbers it means that `NaN` is
equal to itself and `-0.0` is not equal to `0.0`.\\n
*^\\n@SinceKotlin("1.1")\\n@kotlin.internal.LowPriorityInOverloadResolution\\npublic actual infix fun <T>
Array<out T>.contentDeepEquals(other: Array<out T>): Boolean {\\n    return
this.contentDeepEquals(other)\\n}\\n}\\n\\n/**\\n * Returns `true` if the two specified arrays are *deeply* equal to one
another.\\n * i.e. contain the same number of the same elements in the same order.\\n * \\n * The specified arrays are
also considered deeply equal if both are `null`.\\n * \\n * If two corresponding elements are nested arrays, they are
also compared deeply.\\n * If any of arrays contains itself on any nesting level the behavior is undefined.\\n * \\n *
The elements of other types are compared for equality with the [equals][Any.equals] function.\\n * For floating point
numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\\n
*^\\n@SinceKotlin("1.4")\\n@library("arrayDeepEquals")\\npublic actual infix fun <T> Array<out
T>?.contentDeepEquals(other: Array<out T>?): Boolean {\\n    definedExternally\\n}\\n}\\n\\n/**\\n * Returns a hash code

```

based on the contents of this array as if it is [List].\n * Nested arrays are treated as lists too.\n * \n * If any of arrays contains itself on any nesting level the behavior is undefined.\n

```

*\n@SinceKotlin("1.1")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic actual fun <T> Array<out T>.contentDeepHashCode(): Int {\n    return this.contentDeepHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n * Nested arrays are treated as lists too.\n * \n * If any of arrays contains itself on any nesting level the behavior is undefined.\n
```

```

*\n@SinceKotlin("1.4")\n@library("arrayDeepHashCode")\npublic actual fun <T> Array<out T>?.contentDeepHashCode(): Int {\n    definedExternally\n}\n\n/**\n * Returns a string representation of the contents of this array as if it is a [List].\n * Nested arrays are treated as lists too.\n * \n * If any of arrays contains itself on any nesting level that reference\n * is rendered as `"[...]"` to prevent recursion.\n * \n * @sample samples.collections.Arrays.ContentOperations.contentDeepToString\n
```

```

*\n@SinceKotlin("1.1")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic actual fun <T> Array<out T>.contentDeepToString(): String {\n    return this.contentDeepToString()\n}\n\n/**\n * Returns a string representation of the contents of this array as if it is a [List].\n * Nested arrays are treated as lists too.\n * \n * If any of arrays contains itself on any nesting level that reference\n * is rendered as `"[...]"` to prevent recursion.\n * \n * @sample samples.collections.Arrays.ContentOperations.contentDeepToString\n
```

```

*\n@SinceKotlin("1.4")\n@library("arrayDeepToString")\npublic actual fun <T> Array<out T>?.contentDeepToString(): String {\n    definedExternally\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n *\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual infix fun <T> Array<out T>.contentEquals(other: Array<out T>): Boolean {\n    return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n *\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual infix fun ByteArray.contentEquals(other: ByteArray): Boolean {\n    return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n *\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual infix fun ShortArray.contentEquals(other: ShortArray): Boolean {\n    return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n *\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual infix fun IntArray.contentEquals(other: IntArray): Boolean {\n    return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n *\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual infix fun LongArray.contentEquals(other: LongArray): Boolean {\n    return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the

```

same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n

```
*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual infix fun
FloatArray.contentEquals(other: FloatArray): Boolean {\n    return this.contentEquals(other)\n}\n\n/**\n * Returns
`true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the
same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals]
function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual infix fun
DoubleArray.contentEquals(other: DoubleArray): Boolean {\n    return this.contentEquals(other)\n}\n\n/**\n *
Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of
the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals]
function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual infix fun
BooleanArray.contentEquals(other: BooleanArray): Boolean {\n    return this.contentEquals(other)\n}\n\n/**\n *
Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of
the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals]
function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual infix fun
CharArray.contentEquals(other: CharArray): Boolean {\n    return this.contentEquals(other)\n}\n\n\n/**\n * Returns
`true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the
same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals]
function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@SinceKotlin("1.4")\n@library("arrayEquals")\npublic actual infix fun <T> Array<out
T>?.contentEquals(other: Array<out T>?): Boolean {\n    definedExternally\n}\n\n\n/**\n * Returns `true` if the two
specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the
same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating
point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@SinceKotlin("1.4")\n@library("arrayEquals")\npublic actual infix fun ByteArray?.contentEquals(other:
ByteArray?): Boolean {\n    definedExternally\n}\n\n\n/**\n * Returns `true` if the two specified arrays are
*structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n
* The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it
means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@SinceKotlin("1.4")\n@library("arrayEquals")\npublic actual infix fun ShortArray?.contentEquals(other:
ShortArray?): Boolean {\n    definedExternally\n}\n\n\n/**\n * Returns `true` if the two specified arrays are
*structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n
* The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it
means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@SinceKotlin("1.4")\n@library("arrayEquals")\npublic actual infix fun IntArray?.contentEquals(other:
IntArray?): Boolean {\n    definedExternally\n}\n\n\n/**\n * Returns `true` if the two specified arrays are
*structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n
* The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it
means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@SinceKotlin("1.4")\n@library("arrayEquals")\npublic actual infix fun LongArray?.contentEquals(other:
LongArray?): Boolean {\n    definedExternally\n}\n\n\n/**\n * Returns `true` if the two specified arrays are
```

structurally equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n

```

*\n@SinceKotlin("1.4")\n@library("arrayEquals")\npublic actual infix fun FloatArray?.contentEquals(other: FloatArray?): Boolean {\n    definedExternally\n}\n\n/**\n * Returns `true` if the two specified arrays are
    
```

structurally equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n

```

*\n@SinceKotlin("1.4")\n@library("arrayEquals")\npublic actual infix fun DoubleArray?.contentEquals(other: DoubleArray?): Boolean {\n    definedExternally\n}\n\n/**\n * Returns `true` if the two specified arrays are
    
```

structurally equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n

```

*\n@SinceKotlin("1.4")\n@library("arrayEquals")\npublic actual infix fun BooleanArray?.contentEquals(other: BooleanArray?): Boolean {\n    definedExternally\n}\n\n/**\n * Returns `true` if the two specified arrays are
    
```

structurally equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n

```

*\n@SinceKotlin("1.4")\n@library("arrayEquals")\npublic actual infix fun CharArray?.contentEquals(other: CharArray?): Boolean {\n    definedExternally\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n *\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual fun <T> Array<out T>.contentHashCode(): Int {\n    return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n *\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual fun ByteArray.contentHashCode(): Int {\n    return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n *\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual fun ShortArray.contentHashCode(): Int {\n    return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n *\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual fun IntArray.contentHashCode(): Int {\n    return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n *\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual fun LongArray.contentHashCode(): Int {\n    return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n *\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual fun FloatArray.contentHashCode(): Int {\n    return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n *\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual fun DoubleArray.contentHashCode(): Int {\n    return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n *\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual fun BooleanArray.contentHashCode(): Int {\n    return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n *\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual fun CharArray.contentHashCode(): Int {\n    return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n *\n@SinceKotlin("1.4")\n@library("arrayHashCode")\npublic actual
    
```



```

fun <T> Array<out T>.contentHashCode(): Int {\n  definedExternally\n}\n\n/**\n * Returns a hash code based on
the contents of this array as if it is [List].\n */\n@SinceKotlin("1.4")\n@library("arrayHashCode")\npublic actual
fun ByteArray?.contentHashCode(): Int {\n  definedExternally\n}\n\n/**\n * Returns a hash code based on the
contents of this array as if it is [List].\n */\n@SinceKotlin("1.4")\n@library("arrayHashCode")\npublic actual fun
ShortArray?.contentHashCode(): Int {\n  definedExternally\n}\n\n/**\n * Returns a hash code based on the
contents of this array as if it is [List].\n */\n@SinceKotlin("1.4")\n@library("arrayHashCode")\npublic actual fun
IntArray?.contentHashCode(): Int {\n  definedExternally\n}\n\n/**\n * Returns a hash code based on the contents
of this array as if it is [List].\n */\n@SinceKotlin("1.4")\n@library("arrayHashCode")\npublic actual fun
LongArray?.contentHashCode(): Int {\n  definedExternally\n}\n\n/**\n * Returns a hash code based on the
contents of this array as if it is [List].\n */\n@SinceKotlin("1.4")\n@library("arrayHashCode")\npublic actual fun
FloatArray?.contentHashCode(): Int {\n  definedExternally\n}\n\n/**\n * Returns a hash code based on the
contents of this array as if it is [List].\n */\n@SinceKotlin("1.4")\n@library("arrayHashCode")\npublic actual fun
DoubleArray?.contentHashCode(): Int {\n  definedExternally\n}\n\n/**\n * Returns a hash code based on the
contents of this array as if it is [List].\n */\n@SinceKotlin("1.4")\n@library("arrayHashCode")\npublic actual fun
BooleanArray?.contentHashCode(): Int {\n  definedExternally\n}\n\n/**\n * Returns a hash code based on the
contents of this array as if it is [List].\n */\n@SinceKotlin("1.4")\n@library("arrayHashCode")\npublic actual fun
CharArray?.contentHashCode(): Int {\n  definedExternally\n}\n\n/**\n * Returns a string representation of the
contents of the specified array as if it is [List].\n */\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
actual <T> Array<out T>.contentToString(): String {\n  return this.contentToString()\n}\n\n/**\n * Returns a
string representation of the contents of the specified array as if it is [List].\n */\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
actual fun ByteArray.contentToString(): String {\n  return this.contentToString()\n}\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n */\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
actual fun ShortArray.contentToString(): String {\n  return this.contentToString()\n}\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n */\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
actual fun IntArray.contentToString(): String {\n  return this.contentToString()\n}\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n */\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
actual fun LongArray.contentToString(): String {\n  return this.contentToString()\n}\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n */\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
actual fun FloatArray.contentToString(): String {\n  return this.contentToString()\n}\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n */\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
actual fun DoubleArray.contentToString(): String {\n  return this.contentToString()\n}\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n */\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic

```

```

actual fun BooleanArray.contentToString(): String {\n  return this.contentToString()\n}\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
actual fun CharArray.contentToString(): String {\n  return this.contentToString()\n}\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun <T> Array<out T>?.contentToString():
String {\n  definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as
if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun ByteArray?.contentToString(): String
{\n  definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun ShortArray?.contentToString(): String
{\n  definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun IntArray?.contentToString(): String {\n
definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun LongArray?.contentToString(): String
{\n  definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun FloatArray?.contentToString(): String
{\n  definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun DoubleArray?.contentToString(): String
{\n  definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun BooleanArray?.contentToString():
String {\n  definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as
if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun CharArray?.contentToString(): String
{\n  definedExternally\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that
array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it
overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset
the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of
the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this
array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex]
or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * \n * @throws
IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified
[destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the
[destination] array.\n */\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT
_ARGUMENTS")\npublic actual inline fun <T> Array<out T>.copyInto(destination: Array<T>, destinationOffset:
Int = 0, startIndex: Int = 0, endIndex: Int = size): Array<T> {\n  arrayCopy(this, destination, destinationOffset,
startIndex, endIndex)\n  return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination]
array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the
subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n *

```

@param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n

```
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun ByteArray.copyInto(destination: ByteArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ByteArray {\n    arrayCopy(this.unsafeCast<Array<Byte>>(), destination.unsafeCast<Array<Byte>>(), destinationOffset, startIndex, endIndex)\n    return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n
```

```
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun ShortArray.copyInto(destination: ShortArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ShortArray {\n    arrayCopy(this.unsafeCast<Array<Short>>(), destination.unsafeCast<Array<Short>>(), destinationOffset, startIndex, endIndex)\n    return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n
```

```
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun IntArray.copyInto(destination: IntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): IntArray {\n    arrayCopy(this.unsafeCast<Array<Int>>(), destination.unsafeCast<Array<Int>>(), destinationOffset, startIndex, endIndex)\n    return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n
```

```
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun LongArray.copyInto(destination: LongArray, destinationOffset: Int = 0,
```

```

startIndex: Int = 0, endIndex: Int = size): LongArray {
    arrayCopy(this.unsafeCast<Array<Long>>(),
destination.unsafeCast<Array<Long>>(), destinationOffset, startIndex, endIndex)\n    return destination\n}\n\n/**\n
* Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the
same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n *
@param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to
copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n *
@param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this
array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange
doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the
[destination] array indices range.\n * \n * @return the [destination] array.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT
_ARGUMENTS")\npublic actual inline fun FloatArray.copyInto(destination: FloatArray, destinationOffset: Int = 0,
startIndex: Int = 0, endIndex: Int = size): FloatArray {
    arrayCopy(this.unsafeCast<Array<Float>>(),
destination.unsafeCast<Array<Float>>(), destinationOffset, startIndex, endIndex)\n    return destination\n}\n\n/**\n
* Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the
same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n *
@param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to
copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n *
@param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this
array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange
doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the
[destination] array indices range.\n * \n * @return the [destination] array.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT
_ARGUMENTS")\npublic actual inline fun DoubleArray.copyInto(destination: DoubleArray, destinationOffset: Int
= 0, startIndex: Int = 0, endIndex: Int = size): DoubleArray {
    arrayCopy(this.unsafeCast<Array<Double>>(),
destination.unsafeCast<Array<Double>>(), destinationOffset, startIndex, endIndex)\n    return
destination\n}\n\n/**\n
* Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the
same array in the [destination] and even specify the subrange so that it overlaps with the
destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the
[destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy,
0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n *
@throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of
range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the
subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is
out of the [destination] array indices range.\n * \n * @return the [destination] array.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT
_ARGUMENTS")\npublic actual inline fun BooleanArray.copyInto(destination: BooleanArray, destinationOffset:
Int = 0, startIndex: Int = 0, endIndex: Int = size): BooleanArray {
    arrayCopy(this.unsafeCast<Array<Boolean>>(), destination.unsafeCast<Array<Boolean>>(), destinationOffset,
startIndex, endIndex)\n    return destination\n}\n\n/**\n
* Copies this array or its subrange into the [destination]
array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the
subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n *
@param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the
beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the
subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or
[IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex

```

```

> endIndex`.n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array
starting at the specified [destinationOffset],.n * or when that index is out of the [destination] array indices range.\n *
\n * @return the [destination] array.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT
_ARGUMENTS")\npublic actual inline fun CharArray.copyInto(destination: CharArray, destinationOffset: Int = 0,
startIndex: Int = 0, endIndex: Int = size): CharArray {\n    arrayCopy(this.unsafeCast<Array<Char>>(),
destination.unsafeCast<Array<Char>>(), destinationOffset, startIndex, endIndex)\n    return destination\n}\n\n/**\n
* Returns new array which is a copy of the original array.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.copyOf\n *\n@Suppress("ACTUAL_WITHOUT_EXPECT",
"NOTHING_TO_INLINE")\npublic actual inline fun <T> Array<out T>.copyOf(): Array<T> {\n    return
this.asDynamic().slice()\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.copyOf\n *\n@Suppress("NOTHING_TO_INLINE")\npublic
actual inline fun ByteArray.copyOf(): ByteArray {\n    return this.asDynamic().slice()\n}\n\n/**\n * Returns new
array which is a copy of the original array.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.copyOf\n *\n@Suppress("NOTHING_TO_INLINE")\npublic
actual inline fun ShortArray.copyOf(): ShortArray {\n    return this.asDynamic().slice()\n}\n\n/**\n * Returns new
array which is a copy of the original array.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.copyOf\n *\n@Suppress("NOTHING_TO_INLINE")\npublic
actual inline fun IntArray.copyOf(): IntArray {\n    return this.asDynamic().slice()\n}\n\n/**\n * Returns new array
which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n
*\npublic actual fun LongArray.copyOf(): LongArray {\n    return withType("LongArray",
this.asDynamic().slice())\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.copyOf\n *\n@Suppress("NOTHING_TO_INLINE")\npublic
actual inline fun FloatArray.copyOf(): FloatArray {\n    return this.asDynamic().slice()\n}\n\n/**\n * Returns new
array which is a copy of the original array.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.copyOf\n *\n@Suppress("NOTHING_TO_INLINE")\npublic
actual inline fun DoubleArray.copyOf(): DoubleArray {\n    return this.asDynamic().slice()\n}\n\n/**\n * Returns
new array which is a copy of the original array.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.copyOf\n *\npublic actual fun BooleanArray.copyOf():
BooleanArray {\n    return withType("BooleanArray", this.asDynamic().slice())\n}\n\n/**\n * Returns new array
which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n
*\npublic actual fun CharArray.copyOf(): CharArray {\n    return withType("CharArray",
this.asDynamic().slice())\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given
[newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize]
is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater
than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n *\npublic actual fun
ByteArray.copyOf(newSize: Int): ByteArray {\n    require(newSize >= 0) { "Invalid new array size: $newSize." }
\n    return fillFrom(this, ByteArray(newSize))\n}\n\n/**\n * Returns new array which is a copy of the original
array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if
necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the
[newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are
filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n
*\npublic actual fun ShortArray.copyOf(newSize: Int): ShortArray {\n    require(newSize >= 0) { "Invalid new
array size: $newSize." } \n    return fillFrom(this, ShortArray(newSize))\n}\n\n/**\n * Returns new array which is a
copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with
zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated
to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array

```

are filled with zero values.

```

@sample
samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
public actual fun
IntArray.copyOf(newSize: Int): IntArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return fillFrom(this, IntArray(newSize))
}
/**
 * Returns new array which is a copy of the original array,
 * resized to the given [newSize].
 * The copy is either truncated or padded at the end with zero values if necessary.
 * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
 * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
 */
@sample
samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
public actual fun
LongArray.copyOf(newSize: Int): LongArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return withType("LongArray", arrayCopyResize(this, newSize, 0L))
}
/**
 * Returns new array which is a copy of the original array, resized to the given [newSize].
 * The copy is either truncated or padded at the end with zero values if necessary.
 * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
 * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
 */
@sample
samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
public actual fun
FloatArray.copyOf(newSize: Int): FloatArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return fillFrom(this, FloatArray(newSize))
}
/**
 * Returns new array which is a copy of the original array, resized to the given [newSize].
 * The copy is either truncated or padded at the end with zero values if necessary.
 * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
 * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
 */
@sample
samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
public actual fun
DoubleArray.copyOf(newSize: Int): DoubleArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return fillFrom(this, DoubleArray(newSize))
}
/**
 * Returns new array which is a copy of the original array, resized to the given [newSize].
 * The copy is either truncated or padded at the end with `false` values if necessary.
 * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
 * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with `false` values.
 */
@sample
samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
public actual fun
BooleanArray.copyOf(newSize: Int): BooleanArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return withType("BooleanArray", arrayCopyResize(this, newSize, false))
}
/**
 * Returns new array which is a copy of the original array, resized to the given [newSize].
 * The copy is either truncated or padded at the end with null char (`\u0000`) values if necessary.
 * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
 * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with null char (`\u0000`) values.
 */
@sample
samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
public actual fun
CharArray.copyOf(newSize: Int): CharArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return withType("CharArray", fillFrom(this, CharArray(newSize)))
}
/**
 * Returns new array which is a copy of the original array, resized to the given [newSize].
 * The copy is either truncated or padded at the end with `null` values if necessary.
 * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
 * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with `null` values.
 */
@sample
samples.collections.Arrays.CopyOfOperations.resizingCopyOf
public actual fun <T> Array<out T>.copyOf(newSize: Int): Array<T?> {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return arrayCopyResize(this, newSize, null)
}
/**
 * Returns a new array which is a copy of the specified range of the original array.
 * @param fromIndex the start of the range (inclusive) to copy.
 * @param toIndex the end of the range (exclusive) to copy.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */

```

```

*\n@Suppress("ACTUAL_WITHOUT_EXPECT")\npublic actual fun <T> Array<out
T>.copyOfRange(fromIndex: Int, toIndex: Int): Array<T> {\n  AbstractList.checkRangeIndexes(fromIndex,
toIndex, size)\n  return this.asDynamic().slice(fromIndex, toIndex)\n}\n\n/**\n * Returns a new array which is a
copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to
copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException
if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentOutOfRangeException if [fromIndex] is greater than [toIndex].\n */\npublic actual fun
ByteArray.copyOfRange(fromIndex: Int, toIndex: Int): ByteArray {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  return this.asDynamic().slice(fromIndex,
toIndex)\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n *
@param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to
copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the
size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic
actual fun ShortArray.copyOfRange(fromIndex: Int, toIndex: Int): ShortArray {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  return this.asDynamic().slice(fromIndex,
toIndex)\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n *
@param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to
copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the
size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic
actual fun IntArray.copyOfRange(fromIndex: Int, toIndex: Int): IntArray {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  return this.asDynamic().slice(fromIndex,
toIndex)\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n *
@param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to
copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the
size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic
actual fun LongArray.copyOfRange(fromIndex: Int, toIndex: Int): LongArray {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  return withType("LongArray",
this.asDynamic().slice(fromIndex, toIndex))\n}\n\n/**\n * Returns a new array which is a copy of the specified
range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param
toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is
less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if
[fromIndex] is greater than [toIndex].\n */\npublic actual fun FloatArray.copyOfRange(fromIndex: Int, toIndex: Int):
FloatArray {\n  AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  return
this.asDynamic().slice(fromIndex, toIndex)\n}\n\n/**\n * Returns a new array which is a copy of the specified range
of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the
end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n */\npublic actual fun DoubleArray.copyOfRange(fromIndex: Int, toIndex: Int): DoubleArray {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  return this.asDynamic().slice(fromIndex,
toIndex)\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n *
@param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to
copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the
size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic
actual fun BooleanArray.copyOfRange(fromIndex: Int, toIndex: Int): BooleanArray {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  return withType("BooleanArray",
this.asDynamic().slice(fromIndex, toIndex))\n}\n\n/**\n * Returns a new array which is a copy of the specified
range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param
toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is

```

less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n\npublic actual fun CharArray.copyOfRange(fromIndex: Int, toIndex: Int): CharArray {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n return withType("CharArray", this.asDynamic()).slice(fromIndex, toIndex)}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun <T> Array<T>.fill(element: T, fromIndex: Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n nativeFill(element, fromIndex, toIndex)}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun ByteArray.fill(element: Byte, fromIndex: Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n nativeFill(element, fromIndex, toIndex)}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun ShortArray.fill(element: Short, fromIndex: Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n nativeFill(element, fromIndex, toIndex)}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun IntArray.fill(element: Int, fromIndex: Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n nativeFill(element, fromIndex, toIndex)}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun LongArray.fill(element: Long, fromIndex: Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n nativeFill(element, fromIndex, toIndex)}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun FloatArray.fill(element: Float, fromIndex: Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n nativeFill(element, fromIndex, toIndex)}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by


```

default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than
the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun DoubleArray.fill(element: Double, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    nativeFill(element, fromIndex, toIndex);\n}\n\n/**\n
* Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range
(inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by
default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than
the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun BooleanArray.fill(element: Boolean, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    nativeFill(element, fromIndex, toIndex);\n}\n\n/**\n
* Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range
(inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by
default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than
the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun CharArray.fill(element: Char, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    nativeFill(element, fromIndex, toIndex);\n}\n\n/**\n
* Returns an array containing all elements of the original array and then the given [element].\n
*\n@Suppress("ACTUAL_WITHOUT_EXPECT", "NOTHING_TO_INLINE")\npublic actual inline operator
fun <T> Array<out T>.plus(element: T): Array<T> {\n    return
this.asDynamic().concat(arrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the original
array and then the given [element].\n *\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator
fun ByteArray.plus(element: Byte): ByteArray {\n    return plus(byteArrayOf(element))\n}\n\n/**\n * Returns an
array containing all elements of the original array and then the given [element].\n
*\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun ShortArray.plus(element: Short):
ShortArray {\n    return plus(shortArrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the
original array and then the given [element].\n *\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline
operator fun IntArray.plus(element: Int): IntArray {\n    return plus(intArrayOf(element))\n}\n\n/**\n * Returns an
array containing all elements of the original array and then the given [element].\n
*\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun LongArray.plus(element: Long):
LongArray {\n    return plus(longArrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the
original array and then the given [element].\n *\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline
operator fun FloatArray.plus(element: Float): FloatArray {\n    return plus(floatArrayOf(element))\n}\n\n/**\n *
Returns an array containing all elements of the original array and then the given [element].\n
*\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun DoubleArray.plus(element:
Double): DoubleArray {\n    return plus(doubleArrayOf(element))\n}\n\n/**\n * Returns an array containing all
elements of the original array and then the given [element].\n *\n@Suppress("NOTHING_TO_INLINE")\npublic
actual inline operator fun BooleanArray.plus(element: Boolean): BooleanArray {\n    return
plus(booleanArrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the original array and then
the given [element].\n *\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun
CharArray.plus(element: Char): CharArray {\n    return plus(charArrayOf(element))\n}\n\n/**\n * Returns an array
containing all elements of the original array and then all elements of the given [elements] collection.\n
*\n@Suppress("ACTUAL_WITHOUT_EXPECT")\npublic actual operator fun <T> Array<out T>.plus(elements:
Collection<T>): Array<T> {\n    return arrayPlusCollection(this, elements)\n}\n\n/**\n * Returns an array
containing all elements of the original array and then all elements of the given [elements] collection.\n *\npublic
actual operator fun ByteArray.plus(elements: Collection<Byte>): ByteArray {\n    return

```

fillFromCollection(this.copyOf(size + elements.size), this.size, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic actual operator fun ShortArray.plus(elements: Collection<Short>): ShortArray {\n return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic actual operator fun IntArray.plus(elements: Collection<Int>): IntArray {\n return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic actual operator fun LongArray.plus(elements: Collection<Long>): LongArray {\n return arrayPlusCollection(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic actual operator fun FloatArray.plus(elements: Collection<Float>): FloatArray {\n return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic actual operator fun DoubleArray.plus(elements: Collection<Double>): DoubleArray {\n return arrayPlusCollection(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic actual operator fun BooleanArray.plus(elements: Collection<Boolean>): BooleanArray {\n return arrayPlusCollection(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic actual operator fun CharArray.plus(elements: Collection<Char>): CharArray {\n return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress("ACTUAL_WITHOUT_EXPECT", "NOTHING_TO_INLINE")\npublic actual inline operator fun <T> Array<out T>.plus(elements: Array<out T>): Array<T> {\n return this.asDynamic().concat(elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun ByteArray.plus(elements: ByteArray): ByteArray {\n return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun ShortArray.plus(elements: ShortArray): ShortArray {\n return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun IntArray.plus(elements: IntArray): IntArray {\n return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun LongArray.plus(elements: LongArray): LongArray {\n return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun FloatArray.plus(elements: FloatArray): FloatArray {\n return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun DoubleArray.plus(elements: DoubleArray): DoubleArray {\n return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun BooleanArray.plus(elements: BooleanArray): BooleanArray {\n return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun CharArray.plus(elements: CharArray): CharArray {\n return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing

all elements of the original array and then the given [element].\n

```

*\n@Suppress("ACTUAL_WITHOUT_EXPECT", "NOTHING_TO_INLINE")\npublic actual inline fun <T>
Array<out T>.plusElement(element: T): Array<T> {\n    return
this.asDynamic().concat(arrayOf(element))\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample
samples.collections.Arrays.Sorting.sortArray\n *\n@library("primitiveArraySort")\npublic actual fun
IntArray.sort(): Unit {\n    definedExternally\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample
samples.collections.Arrays.Sorting.sortArray\n *\npublic actual fun LongArray.sort(): Unit {\n
@Suppress("DEPRECATION")\n    if (size > 1) sort { a: Long, b: Long -> a.compareTo(b) }\n}\n\n/**\n * Sorts
the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n
*\n@library("primitiveArraySort")\npublic actual fun ByteArray.sort(): Unit {\n    definedExternally\n}\n\n/**\n
 * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n
*\n@library("primitiveArraySort")\npublic actual fun ShortArray.sort(): Unit {\n    definedExternally\n}\n\n/**\n
 * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n
*\n@library("primitiveArraySort")\npublic actual fun DoubleArray.sort(): Unit {\n
definedExternally\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample
samples.collections.Arrays.Sorting.sortArray\n *\n@library("primitiveArraySort")\npublic actual fun
FloatArray.sort(): Unit {\n    definedExternally\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample
samples.collections.Arrays.Sorting.sortArray\n *\n@library("primitiveArraySort")\npublic actual fun
CharArray.sort(): Unit {\n    definedExternally\n}\n\n/**\n * Sorts the array in-place according to the natural order
of its elements.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other
after sorting.\n * \n * @sample samples.collections.Arrays.Sorting.sortArrayOfComparable\n *\npublic actual fun
<T : Comparable<T>> Array<out T>.sort(): Unit {\n    if (size > 1) sortArray(this)\n}\n\n/**\n * Sorts the array in-
place according to the order specified by the given [comparison] function.\n * \n * The sort is _stable_. It means that
equal elements preserve their order relative to each other after sorting.\n * \n *@Deprecated("Use sortWith instead",
ReplaceWith("this.sortWith(Comparator(comparison))"))\n@DeprecatedSinceKotlin(warningSince =
"1.6")\npublic fun <T> Array<out T>.sort(comparison: (a: T, b: T) -> Int): Unit {\n    if (size > 1)
sortArrayWith(this, comparison)\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * The sort is _stable_. It
means that equal elements preserve their order relative to each other after sorting.\n * \n * @param fromIndex the
start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size
of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex]
is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than
[toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArrayOfComparable\n
*\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun <T : Comparable<T>> Array<out T>.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    sortArrayWith(this, fromIndex, toIndex,
naturalOrder())\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range
(inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by
default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than
the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n *
@sample samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun ByteArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    val subarray =
this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<ByteArray>()\n    subarray.sort()\n}\n\n/**\n * Sorts a
range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n *
@param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample

```

```

samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun ShortArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    val subarray =
this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<ShortArray>()\n    subarray.sort()\n}\n\n/**\n * Sorts a
range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n *
@param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample
samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun IntArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    val subarray =
this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<IntArray>()\n    subarray.sort()\n}\n\n/**\n * Sorts a
range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n *
@param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample
samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun LongArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    sortArrayWith(this.unsafeCast<Array<Long>>(),
fromIndex, toIndex, naturalOrder())\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex
the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort,
size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or
[toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than
[toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun FloatArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    val subarray =
this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<FloatArray>()\n    subarray.sort()\n}\n\n/**\n * Sorts a
range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n *
@param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample
samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun DoubleArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    val subarray =
this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<DoubleArray>()\n    subarray.sort()\n}\n\n/**\n * Sorts a
range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n *
@param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample
samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun CharArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    val subarray =

```

```

this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<CharArray>()\n  subarray.sort()\n}\n\n/**\n * Sorts the
array in-place according to the order specified by the given [comparison] function.\n */\n@Deprecated("Use other
sorting functions from the Standard Library")\n@DeprecatedSinceKotlin(warningSince =
"1.6")\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.sort(noinline comparison: (a: Byte, b: Byte) ->
Int): Unit {\n  nativeSort(comparison)\n}\n\n/**\n * Sorts the array in-place according to the order specified by the
given [comparison] function.\n */\n@Deprecated("Use other sorting functions from the Standard
Library")\n@DeprecatedSinceKotlin(warningSince = "1.6")\n@kotlin.internal.InlineOnly\npublic inline fun
ShortArray.sort(noinline comparison: (a: Short, b: Short) -> Int): Unit {\n  nativeSort(comparison)\n}\n\n/**\n *
Sorts the array in-place according to the order specified by the given [comparison] function.\n
*/\n@Deprecated("Use other sorting functions from the Standard
Library")\n@DeprecatedSinceKotlin(warningSince = "1.6")\n@kotlin.internal.InlineOnly\npublic inline fun
IntArray.sort(noinline comparison: (a: Int, b: Int) -> Int): Unit {\n  nativeSort(comparison)\n}\n\n/**\n * Sorts the
array in-place according to the order specified by the given [comparison] function.\n */\n@Deprecated("Use other
sorting functions from the Standard Library")\n@DeprecatedSinceKotlin(warningSince =
"1.6")\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.sort(noinline comparison: (a: Long, b: Long) ->
Int): Unit {\n  nativeSort(comparison)\n}\n\n/**\n * Sorts the array in-place according to the order specified by the
given [comparison] function.\n */\n@Deprecated("Use other sorting functions from the Standard
Library")\n@DeprecatedSinceKotlin(warningSince = "1.6")\n@kotlin.internal.InlineOnly\npublic inline fun
FloatArray.sort(noinline comparison: (a: Float, b: Float) -> Int): Unit {\n  nativeSort(comparison)\n}\n\n/**\n *
Sorts the array in-place according to the order specified by the given [comparison] function.\n
*/\n@Deprecated("Use other sorting functions from the Standard
Library")\n@DeprecatedSinceKotlin(warningSince = "1.6")\n@kotlin.internal.InlineOnly\npublic inline fun
DoubleArray.sort(noinline comparison: (a: Double, b: Double) -> Int): Unit {\n
  nativeSort(comparison)\n}\n\n/**\n * Sorts the array in-place according to the order specified by the given
[comparison] function.\n */\n@Deprecated("Use other sorting functions from the Standard
Library")\n@DeprecatedSinceKotlin(warningSince = "1.6")\n@kotlin.internal.InlineOnly\npublic inline fun
CharArray.sort(noinline comparison: (a: Char, b: Char) -> Int): Unit {\n  nativeSort(comparison)\n}\n\n/**\n *
Sorts the array in-place according to the order specified by the given [comparator].\n * \n * The sort is _stable_. It
means that equal elements preserve their order relative to each other after sorting.\n */\npublic actual fun <T>
Array<out T>.sortWith(comparator: Comparator<in T>): Unit {\n  if (size > 1) sortArrayWith(this,
comparator)\n}\n\n/**\n * Sorts a range in the array in-place with the given [comparator].\n * \n * The sort is
_stable_. It means that equal elements preserve their order relative to each other after sorting.\n * \n * @param
fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range
(exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is
less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if
[fromIndex] is greater than [toIndex].\n
*/\n@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun <T> Array<out T>.sortWith(comparator: Comparator<in T>, fromIndex: Int = 0, toIndex: Int = size):
Unit {\n  AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  sortArrayWith(this, fromIndex, toIndex,
comparator)\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n
*/\npublic actual fun ByteArray.toTypedArray(): Array<Byte> {\n  return js("[]").slice.call(this)\n}\n\n/**\n *
Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic actual fun
ShortArray.toTypedArray(): Array<Short> {\n  return js("[]").slice.call(this)\n}\n\n/**\n * Returns a *typed*
object array containing all of the elements of this primitive array.\n */\npublic actual fun IntArray.toTypedArray():
Array<Int> {\n  return js("[]").slice.call(this)\n}\n\n/**\n * Returns a *typed* object array containing all of the
elements of this primitive array.\n */\npublic actual fun LongArray.toTypedArray(): Array<Long> {\n  return
js("[]").slice.call(this)\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive
array.\n */\npublic actual fun FloatArray.toTypedArray(): Array<Float> {\n  return

```

```

js("[\"]).slice.call(this)\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive
array.\n */\npublic actual fun DoubleArray.toArray(): Array<Double> {\n    return
js("[\"]).slice.call(this)\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive
array.\n */\npublic actual fun BooleanArray.toArray(): Array<Boolean> {\n    return
js("[\"]).slice.call(this)\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive
array.\n */\npublic actual fun CharArray.toArray(): Array<Char> {\n    return Array(size) { index ->
this[index] }\n}\n\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*/\n@file:kotlin.jvm.JvmName("ComparisonsKt")\n@file:kotlin.jvm.JvmMultifileClass\n\npackage
kotlin.comparisons\n\n/**\n * Compares two values using the specified functions [selectors] to calculate the result
of the comparison.\n * The functions are called sequentially, receive the given values [a] and [b] and return
[Comparable]\n * objects. As soon as the [Comparable] instances returned by a function for [a] and [b] values do
not\n * compare as equal, the result of that comparison is returned.\n */\n * @sample
samples.comparisons.Comparisons.compareValuesByWithSelectors\n */\npublic fun <T> compareValuesBy(a: T,
b: T, vararg selectors: (T) -> Comparable<*>?): Int {\n    require(selectors.size > 0)\n    return
compareValuesByImpl(a, b, selectors)\n}\n\nprivate fun <T> compareValuesByImpl(a: T, b: T, selectors:
Array<out (T) -> Comparable<*>?): Int {\n    for (fn in selectors) {\n        val v1 = fn(a)\n        val v2 = fn(b)\n        val diff = compareValues(v1, v2)\n        if (diff != 0) return diff\n    }\n    return 0\n}\n\n/**\n * Compares two
values using the specified [selector] function to calculate the result of the comparison.\n * The function is applied to
the given values [a] and [b] and return [Comparable] objects.\n * The result of comparison of these [Comparable]
instances is returned.\n */\n * @sample samples.comparisons.Comparisons.compareValuesByWithSingleSelector\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun <T> compareValuesBy(a: T, b: T, selector: (T) ->
Comparable<*>?): Int {\n    return compareValues(selector(a), selector(b))\n}\n\n/**\n * Compares two values
using the specified [selector] function to calculate the result of the comparison.\n * The function is applied to the
given values [a] and [b] and return objects of type K which are then being\n * compared with the given
[comparator].\n */\n * @sample samples.comparisons.Comparisons.compareValuesByWithComparator\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun <T, K> compareValuesBy(a: T, b: T, comparator: Comparator<in
K>, selector: (T) -> K): Int {\n    return comparator.compare(selector(a), selector(b))\n}\n\n//// Not so useful without
type inference for receiver of expression\n//// compareValuesWith(v1, v2, compareBy { it.prop1 }
thenByDescending { it.prop2 })\n\n/**\n */\n * Compares two values using the specified [comparator].\n */\n
*/\n@Suppress("NOTHING_TO_INLINE")\npublic inline fun <T> compareValuesWith(a: T, b: T, comparator:
Comparator<T>): Int = comparator.compare(a, b)\n\n/**\n * Compares two nullable [Comparable] values. Null
is considered less than any value.\n */\n * @sample samples.comparisons.Comparisons.compareValues\n */\npublic
fun <T : Comparable<*>> compareValues(a: T?, b: T?): Int {\n    if (a === b) return 0\n    if (a == null) return -1\n
if (b == null) return 1\n    @Suppress("UNCHECKED_CAST")\n    return (a as
Comparable<Any>).compareTo(b)\n}\n\n/**\n * Creates a comparator using the sequence of functions to calculate a
result of comparison.\n * The functions are called sequentially, receive the given values `a` and `b` and return
[Comparable]\n * objects. As soon as the [Comparable] instances returned by a function for `a` and `b` values do
not\n * compare as equal, the result of that comparison is returned from the [Comparator].\n */\n * @sample
samples.comparisons.Comparisons.compareByWithSelectors\n */\npublic fun <T> compareBy(vararg selectors: (T)
-> Comparable<*>?): Comparator<T> {\n    require(selectors.size > 0)\n    return Comparator { a, b ->
compareValuesByImpl(a, b, selectors) }\n}\n\n/**\n * Creates a comparator using the function to transform value
to a [Comparable] instance for comparison.\n */\n * @sample
samples.comparisons.Comparisons.compareByWithSingleSelector\n */\n@kotlin.internal.InlineOnly\npublic inline
fun <T> compareBy(crossinline selector: (T) -> Comparable<*>?): Comparator<T> =\n    Comparator { a, b ->
compareValuesBy(a, b, selector) }\n\n/**\n * Creates a comparator using the [selector] function to transform values
being compared and then applying\n * the specified [comparator] to compare transformed values.\n */\n * @sample

```

```

samples.comparisons.Comparisons.compareByWithComparator\n *\n@kotlin.internal.InlineOnly\npublic inline
fun <T, K> compareBy(comparator: Comparator<in K>, crossinline selector: (T) -> K): Comparator<T> =\n
Comparator { a, b -> compareValuesBy(a, b, comparator, selector) }\n\n/**\n * Creates a descending comparator
using the function to transform value to a [Comparable] instance for comparison.\n *\n * @sample
samples.comparisons.Comparisons.compareByDescendingWithSingleSelector\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T> compareByDescending(crossinline selector: (T) ->
Comparable<*>?): Comparator<T> =\n  Comparator { a, b -> compareValuesBy(b, a, selector) }\n\n/**\n *
Creates a descending comparator using the [selector] function to transform values being compared and then
applying\n * the specified [comparator] to compare transformed values.\n *\n * Note that an order of [comparator] is
reversed by this wrapper.\n *\n * @sample
samples.comparisons.Comparisons.compareByDescendingWithComparator\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T, K> compareByDescending(comparator: Comparator<in K>,
crossinline selector: (T) -> K): Comparator<T> =\n  Comparator { a, b -> compareValuesBy(b, a, comparator,
selector) }\n\n/**\n * Creates a comparator comparing values after the primary comparator defined them equal. It
uses\n * the function to transform value to a [Comparable] instance for comparison.\n *\n * @sample
samples.comparisons.Comparisons.thenBy\n *\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Comparator<T>.thenBy(crossinline selector: (T) -> Comparable<*>?): Comparator<T> =\n  Comparator { a, b -
>\n    val previousCompare = this@thenBy.compare(a, b)\n    if (previousCompare != 0) previousCompare else
compareValuesBy(a, b, selector)\n  }\n\n/**\n * Creates a comparator comparing values after the primary
comparator defined them equal. It uses\n * the [selector] function to transform values and then compares them with
the given [comparator].\n *\n * @sample
samples.comparisons.Comparisons.thenByWithComparator\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T, K> Comparator<T>.thenBy(comparator: Comparator<in K>,
crossinline selector: (T) -> K): Comparator<T> =\n  Comparator { a, b ->\n    val previousCompare =
this@thenBy.compare(a, b)\n    if (previousCompare != 0) previousCompare else compareValuesBy(a, b,
comparator, selector)\n  }\n\n/**\n * Creates a descending comparator using the primary comparator and\n * the
function to transform value to a [Comparable] instance for comparison.\n *\n * @sample
samples.comparisons.Comparisons.thenByDescending\n *\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Comparator<T>.thenByDescending(crossinline selector: (T) -> Comparable<*>?): Comparator<T> =\n
Comparator { a, b ->\n    val previousCompare = this@thenByDescending.compare(a, b)\n    if
(previousCompare != 0) previousCompare else compareValuesBy(b, a, selector)\n  }\n\n/**\n * Creates a
descending comparator comparing values after the primary comparator defined them equal. It uses\n * the [selector]
function to transform values and then compares them with the given [comparator].\n *\n * @sample
samples.comparisons.Comparisons.thenByDescendingWithComparator\n *\n@kotlin.internal.InlineOnly\npublic
inline fun <T, K> Comparator<T>.thenByDescending(comparator: Comparator<in K>, crossinline selector: (T) ->
K): Comparator<T> =\n  Comparator { a, b ->\n    val previousCompare = this@thenByDescending.compare(a,
b)\n    if (previousCompare != 0) previousCompare else compareValuesBy(b, a, comparator, selector)\n  }\n\n
**\n * Creates a comparator using the primary comparator and function to calculate a result of comparison.\n
*\n * @sample
samples.comparisons.Comparisons.thenComparator\n *\n@kotlin.internal.InlineOnly\npublic inline
fun <T> Comparator<T>.thenComparator(crossinline comparison: (a: T, b: T) -> Int): Comparator<T> =\n
Comparator { a, b ->\n    val previousCompare = this@thenComparator.compare(a, b)\n    if (previousCompare
!= 0) previousCompare else comparison(a, b)\n  }\n\n/**\n * Combines this comparator and the given [comparator]
such that the latter is applied only\n * when the former considered values equal.\n *\n * @sample
samples.comparisons.Comparisons.then\n *\npublic infix fun <T> Comparator<T>.then(comparator:
Comparator<in T>): Comparator<T> =\n  Comparator { a, b ->\n    val previousCompare =
this@then.compare(a, b)\n    if (previousCompare != 0) previousCompare else comparator.compare(a, b)\n  }\n\n
**\n * Combines this comparator and the given [comparator] such that the latter is applied only\n * when the
former considered values equal.\n *\n * @sample
samples.comparisons.Comparisons.thenDescending\n *\npublic infix fun <T> Comparator<T>.thenDescending(comparator: Comparator<in T>): Comparator<T> =\n

```

```

Comparator<T> { a, b ->\n    val previousCompare = this@thenDescending.compare(a, b)\n    if
(previousCompare != 0) previousCompare else comparator.compare(b, a)\n    }\n\n// Not so useful without type
inference for receiver of expression\n/**\n * Extends the given [comparator] of non-nullable values to a comparator
of nullable values\n * considering `null` value less than any other value.\n *\n * @sample
samples.comparisons.Comparisons.nullsFirstLastWithComparator\n *\npublic fun <T : Any>
nullsFirst(comparator: Comparator<in T>): Comparator<T?> =\n    Comparator { a, b ->\n        when {\n            a
=== b -> 0\n            a == null -> -1\n            b == null -> 1\n            else -> comparator.compare(a, b)\n        }\n    }\n\n/**\n * Provides a comparator of nullable [Comparable] values\n * considering `null` value less than any other
value.\n *\n * @sample samples.comparisons.Comparisons.nullsFirstLastComparator\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T : Comparable<T>> nullsFirst(): Comparator<T?> =
nullsFirst(naturalOrder())\n\n/**\n * Extends the given [comparator] of non-nullable values to a comparator of
nullable values\n * considering `null` value greater than any other value.\n *\n * @sample
samples.comparisons.Comparisons.nullsFirstLastWithComparator\n *\npublic fun <T : Any>
nullsLast(comparator: Comparator<in T>): Comparator<T?> =\n    Comparator { a, b ->\n        when {\n            a
=== b -> 0\n            a == null -> 1\n            b == null -> -1\n            else -> comparator.compare(a, b)\n        }\n    }\n\n/**\n * Provides a comparator of nullable [Comparable] values\n * considering `null` value greater than any
other value.\n *\n * @sample samples.comparisons.Comparisons.nullsFirstLastComparator\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T : Comparable<T>> nullsLast(): Comparator<T?> =
nullsLast(naturalOrder())\n\n/**\n * Returns a comparator that compares [Comparable] objects in natural order.\n
*\n * @sample samples.comparisons.Comparisons.naturalOrderComparator\n *\npublic fun <T : Comparable<T>>
naturalOrder(): Comparator<T> = @Suppress("UNCHECKED_CAST") (NaturalOrderComparator as
Comparator<T>)\n\n/**\n * Returns a comparator that compares [Comparable] objects in reversed natural order.\n
*\n * @sample samples.comparisons.Comparisons.nullsFirstLastWithComparator\n *\npublic fun <T :
Comparable<T>> reverseOrder(): Comparator<T> = @Suppress("UNCHECKED_CAST")
(ReverseOrderComparator as Comparator<T>)\n\n/**\n * Returns a comparator that imposes the reverse ordering
of this comparator.\n *\n * @sample samples.comparisons.Comparisons.reversed\n
*\n@Suppress("EXTENSION_SHADOWED_BY_MEMBER")\npublic fun <T> Comparator<T>.reversed():
Comparator<T> = when (this) {\n    is ReversedComparator -> this.comparator\n    NaturalOrderComparator ->
@Suppress("UNCHECKED_CAST") (ReverseOrderComparator as Comparator<T>)\n    ReverseOrderComparator -> @Suppress("UNCHECKED_CAST") (NaturalOrderComparator as
Comparator<T>)\n    else -> ReversedComparator(this)\n}\n\nprivate class ReversedComparator<T>(public val
comparator: Comparator<T>) : Comparator<T> {\n    override fun compare(a: T, b: T): Int = comparator.compare(b,
a)\n    @Suppress("VIRTUAL_MEMBER_HIDDEN")\n    fun reversed(): Comparator<T> =
comparator\n}\n\nprivate object NaturalOrderComparator : Comparator<Comparable<Any>> {\n    override fun
compare(a: Comparable<Any>, b: Comparable<Any>): Int = a.compareTo(b)\n}\n\n@Suppress("VIRTUAL_MEMBER_HIDDEN")\nprivate object ReverseOrderComparator : Comparator<Comparable<Any>> {\n    override fun
compare(a: Comparable<Any>, b: Comparable<Any>): Int = b.compareTo(a)\n}\n\n@Suppress("VIRTUAL_MEMBER_HIDDEN")\nprivate object NaturalOrderComparator\n}\n\n"/**\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("StandardKt")\npackage kotlin\n\nimport
kotlin.contracts.*\n\n/**\n * An exception is thrown to indicate that a method body remains to be implemented.\n
*\n * @public class NotImplementedError(message: String = "An operation is not implemented.") :
Error(message)\n\n/**\n * Always throws [NotImplementedError] stating that operation is not implemented.\n
*\n * @kotlin.internal.InlineOnly\npublic inline fun TODO(): Nothing = throw NotImplementedError()\n\n/**\n * Always throws [NotImplementedError] stating that operation is not implemented.\n *\n * @param reason a string

```


explaining why the implementation is missing.

```

*\n@kotlin.internal.InlineOnly\npublic inline fun TODO(reason:
String): Nothing = throw NotImplementedError("An operation is not implemented: $reason")\n\n\n/**\n * Calls
the specified function [block] and returns its result.\n *\n * For detailed usage information see the documentation for
[scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#run).\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <R> run(block: () -> R): R {\n  contract {\n
callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n  }\n  return block()\n}\n\n\n/**\n * Calls the specified
function [block] with `this` value as its receiver and returns its result.\n *\n * For detailed usage information see the
documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#run).\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> T.run(block: T.() -> R): R {\n  contract {\n
callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n  }\n  return block()\n}\n\n\n/**\n * Calls the specified
function [block] with the given [receiver] as its receiver and returns its result.\n *\n * For detailed usage information
see the documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#with).\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> with(receiver: T, block: T.() -> R): R {\n  contract {\n
callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n  }\n  return receiver.block()\n}\n\n\n/**\n * Calls the
specified function [block] with `this` value as its receiver and returns `this` value.\n *\n * For detailed usage
information see the documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-
functions.html#apply).\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T> T.apply(block: T.() -> Unit): T {\n  contract {\n
callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n  }\n  block()\n  return
this\n}\n\n\n/**\n * Calls the specified function [block] with `this` value as its argument and returns `this` value.\n *\n
* For detailed usage information see the documentation for [scope
functions](https://kotlinlang.org/docs/reference/scope-functions.html#also).\n
*\n@kotlin.internal.InlineOnly\n@SinceKotlin("1.1")\npublic inline fun <T> T.also(block: (T) -> Unit): T {\n
contract {\n  callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n  }\n  block(this)\n  return
this\n}\n\n\n/**\n * Calls the specified function [block] with `this` value as its argument and returns its result.\n *\n
* For detailed usage information see the documentation for [scope
functions](https://kotlinlang.org/docs/reference/scope-functions.html#let).\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> T.let(block: (T) -> R): R {\n  contract {\n
callsInPlace(block,
InvocationKind.EXACTLY_ONCE)\n  }\n  return block(this)\n}\n\n\n/**\n * Returns `this` value if it satisfies the
given [predicate] or `null`, if it doesn't.\n *\n * For detailed usage information see the documentation for [scope
functions](https://kotlinlang.org/docs/reference/scope-functions.html#takeif-and-takeunless).\n
*\n@kotlin.internal.InlineOnly\n@SinceKotlin("1.1")\npublic inline fun <T> T.takeIf(predicate: (T) -> Boolean):
T? {\n  contract {\n  callsInPlace(predicate, InvocationKind.EXACTLY_ONCE)\n  }\n  return if
(predicate(this)) this else null\n}\n\n\n/**\n * Returns `this` value if it _does not_ satisfy the given [predicate] or
`null`, if it does.\n *\n * For detailed usage information see the documentation for [scope
functions](https://kotlinlang.org/docs/reference/scope-functions.html#takeif-and-takeunless).\n
*\n@kotlin.internal.InlineOnly\n@SinceKotlin("1.1")\npublic inline fun <T> T.takeUnless(predicate: (T) ->
Boolean): T? {\n  contract {\n  callsInPlace(predicate, InvocationKind.EXACTLY_ONCE)\n  }\n  return if
(!predicate(this)) this else null\n}\n\n\n/**\n * Executes the given function [action] specified number of [times].\n *\n
* A zero-based index of current iteration is passed as a parameter to [action].\n *\n * @sample
samples.misc.ControlFlow.repeat\n
*\n@kotlin.internal.InlineOnly\npublic inline fun repeat(times: Int, action: (Int)
-> Unit) {\n  contract { callsInPlace(action) }\n  for (index in 0 until times) {\n    action(index)\n
}\n}\n\n", /*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n@npackage kotlin.comparisons\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the
GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n\nimport
kotlin.js.*\n\n\n/**\n * Returns the greater of two values.\n *\n * If values are equal, returns the first one.\n
*\n@kotlin.internal.InlineOnly\n@SinceKotlin("1.1")\npublic actual fun <T : Comparable<T>> maxOf(a: T, b: T): T {\n
return if (a >= b) a
else b\n}\n\n\n/**\n * Returns the greater of two values.\n

```

```

*^@SinceKotlin("1.1")^@kotlin.internal.InlineOnly^@public actual inline fun maxOf(a: Byte, b: Byte): Byte {^n
    return maxOf(a.toInt(), b.toInt()).unsafeCast<Byte>()^n}^n/**^n * Returns the greater of two values.^n
*^@SinceKotlin("1.1")^@kotlin.internal.InlineOnly^@public actual inline fun maxOf(a: Short, b: Short): Short
{^n    return maxOf(a.toInt(), b.toInt()).unsafeCast<Short>()^n}^n/**^n * Returns the greater of two values.^n
*^@SinceKotlin("1.1")^@kotlin.internal.InlineOnly^@public actual inline fun maxOf(a: Int, b: Int): Int {^n
    return JsMath.max(a, b)^n}^n/**^n * Returns the greater of two values.^n
*^@SinceKotlin("1.1")^@Suppress("NOTHING_TO_INLINE")^@public actual inline fun maxOf(a: Long, b:
Long): Long {^n    return if (a >= b) a else b^nn}^n/**^n * Returns the greater of two values.^n * ^n * If either value
is `NaN`, returns `NaN`.^n
*^@SinceKotlin("1.1")^@kotlin.internal.InlineOnly^@public actual inline fun
maxOf(a: Float, b: Float): Float {^n    return JsMath.max(a, b)^n}^n/**^n * Returns the greater of two values.^n * ^n
* If either value is `NaN`, returns `NaN`.^n
*^@SinceKotlin("1.1")^@kotlin.internal.InlineOnly^@public actual
inline fun maxOf(a: Double, b: Double): Double {^n    return JsMath.max(a, b)^n}^n/**^n * Returns the greater of
three values.^n * ^n * If there are multiple equal maximal values, returns the first of them.^n
*^@SinceKotlin("1.1")^@public actual fun <T : Comparable<T>> maxOf(a: T, b: T, c: T): T {^n    return
maxOf(a, maxOf(b, c))^n}^n/**^n * Returns the greater of three values.^n
*^@SinceKotlin("1.1")^@kotlin.internal.InlineOnly^@public actual inline fun maxOf(a: Byte, b: Byte, c: Byte):
Byte {^n    return JsMath.max(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Byte>()^n}^n/**^n * Returns the greater of
three values.^n
*^@SinceKotlin("1.1")^@kotlin.internal.InlineOnly^@public actual inline fun maxOf(a: Short, b:
Short, c: Short): Short {^n    return JsMath.max(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Short>()^n}^n/**^n *
Returns the greater of three values.^n
*^@SinceKotlin("1.1")^@kotlin.internal.InlineOnly^@public actual inline
fun maxOf(a: Int, b: Int, c: Int): Int {^n    return JsMath.max(a, b, c)^n}^n/**^n * Returns the greater of three
values.^n
*^@SinceKotlin("1.1")^@kotlin.internal.InlineOnly^@public actual inline fun maxOf(a: Long, b: Long,
c: Long): Long {^n    return maxOf(a, maxOf(b, c))^n}^n/**^n * Returns the greater of three values.^n * ^n * If any
value is `NaN`, returns `NaN`.^n
*^@SinceKotlin("1.1")^@kotlin.internal.InlineOnly^@public actual inline fun
maxOf(a: Float, b: Float, c: Float): Float {^n    return JsMath.max(a, b, c)^n}^n/**^n * Returns the greater of three
values.^n * ^n * If any value is `NaN`, returns `NaN`.^n
*^@SinceKotlin("1.1")^@kotlin.internal.InlineOnly^@public actual inline fun maxOf(a: Double, b: Double, c:
Double): Double {^n    return JsMath.max(a, b, c)^n}^n/**^n * Returns the greater of the given values.^n * ^n * If
there are multiple equal maximal values, returns the first of them.^n
*^@SinceKotlin("1.4")^@public actual fun <T
: Comparable<T>> maxOf(a: T, vararg other: T): T {^n    var max = a^nn    for (e in other) max = maxOf(max, e)^nn
return max^nn}^n/**^n * Returns the greater of the given values.^n
*^@SinceKotlin("1.4")^@public actual fun
maxOf(a: Byte, vararg other: Byte): Byte {^n    var max = a^nn    for (e in other) max = maxOf(max, e)^nn    return
max^nn}^n/**^n * Returns the greater of the given values.^n
*^@SinceKotlin("1.4")^@public actual fun maxOf(a:
Short, vararg other: Short): Short {^n    var max = a^nn    for (e in other) max = maxOf(max, e)^nn    return
max^nn}^n/**^n * Returns the greater of the given values.^n
*^@SinceKotlin("1.4")^@public actual fun maxOf(a:
Int, vararg other: Int): Int {^n    var max = a^nn    for (e in other) max = maxOf(max, e)^nn    return max^nn}^n
/**^n * Returns the greater of the given values.^n
*^@SinceKotlin("1.4")^@public actual fun maxOf(a: Long, vararg
other: Long): Long {^n    var max = a^nn    for (e in other) max = maxOf(max, e)^nn    return max^nn}^n/**^n *
Returns the greater of the given values.^n * ^n * If any value is `NaN`, returns `NaN`.^n
*^@SinceKotlin("1.4")^@public actual fun maxOf(a: Float, vararg other: Float): Float {^n    var max = a^nn    for (e
in other) max = maxOf(max, e)^nn    return max^nn}^n/**^n * Returns the greater of the given values.^n * ^n * If any
value is `NaN`, returns `NaN`.^n
*^@SinceKotlin("1.4")^@public actual fun maxOf(a: Double, vararg other:
Double): Double {^n    var max = a^nn    for (e in other) max = maxOf(max, e)^nn    return max^nn}^n/**^n * Returns
the smaller of two values.^n * ^n * If values are equal, returns the first one.^n
*^@SinceKotlin("1.1")^@public
actual fun <T : Comparable<T>> minOf(a: T, b: T): T {^n    return if (a <= b) a else b^nn}^n/**^n * Returns the
smaller of two values.^n
*^@SinceKotlin("1.1")^@kotlin.internal.InlineOnly^@public actual inline fun minOf(a:
Byte, b: Byte): Byte {^n    return minOf(a.toInt(), b.toInt()).unsafeCast<Byte>()^n}^n/**^n * Returns the smaller of
two values.^n
*^@SinceKotlin("1.1")^@kotlin.internal.InlineOnly^@public actual inline fun minOf(a: Short, b:

```

```

Short): Short { \n    return minOf(a.toInt(), b.toInt()).unsafeCast<Short>()\n}\n\n/**\n * Returns the smaller of two values.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Int, b: Int): Int\n{\n    return JsMath.min(a, b)\n}\n\n/**\n * Returns the smaller of two values.\n */\n@SinceKotlin("1.1")\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline fun minOf(a: Long, b:\nLong): Long {\n    return if (a <= b) a else b\n}\n\n/**\n * Returns the smaller of two values.\n * \n * If either value\nis `NaN`, returns `NaN`.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun\nminOf(a: Float, b: Float): Float {\n    return JsMath.min(a, b)\n}\n\n/**\n * Returns the smaller of two values.\n * \n * If either value is `NaN`, returns `NaN`.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual\ninline fun minOf(a: Double, b: Double): Double {\n    return JsMath.min(a, b)\n}\n\n/**\n * Returns the smaller of\nthree values.\n * \n * If there are multiple equal minimal values, returns the first of them.\n */\n@SinceKotlin("1.1")\npublic actual fun <T : Comparable<T>> minOf(a: T, b: T, c: T): T {\n    return minOf(a,\nminOf(b, c))\n}\n\n/**\n * Returns the smaller of three values.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Byte, b: Byte, c: Byte):\nByte {\n    return JsMath.min(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Byte>()\n}\n\n/**\n * Returns the smaller of\nthree values.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Short, b:\nShort, c: Short): Short {\n    return JsMath.min(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Short>()\n}\n\n/**\n * Returns the smaller of three values.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline\nfun minOf(a: Int, b: Int, c: Int): Int {\n    return JsMath.min(a, b, c)\n}\n\n/**\n * Returns the smaller of three\nvalues.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Long, b: Long,\nc: Long): Long {\n    return minOf(a, minOf(b, c))\n}\n\n/**\n * Returns the smaller of three values.\n * \n * If any\nvalue is `NaN`, returns `NaN`.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun\nminOf(a: Float, b: Float, c: Float): Float {\n    return JsMath.min(a, b, c)\n}\n\n/**\n * Returns the smaller of three\nvalues.\n * \n * If any value is `NaN`, returns `NaN`.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Double, b: Double, c:\nDouble): Double {\n    return JsMath.min(a, b, c)\n}\n\n/**\n * Returns the smaller of the given values.\n * \n * If\nthere are multiple equal minimal values, returns the first of them.\n */\n@SinceKotlin("1.4")\npublic actual fun <T\n: Comparable<T>> minOf(a: T, vararg other: T): T {\n    var min = a\n    for (e in other) min = minOf(min, e)\n    return min\n}\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin("1.4")\npublic actual fun\nminOf(a: Byte, vararg other: Byte): Byte {\n    var min = a\n    for (e in other) min = minOf(min, e)\n    return\nmin\n}\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin("1.4")\npublic actual fun minOf(a:\nShort, vararg other: Short): Short {\n    var min = a\n    for (e in other) min = minOf(min, e)\n    return\nmin\n}\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin("1.4")\npublic actual fun minOf(a:\nInt, vararg other: Int): Int {\n    var min = a\n    for (e in other) min = minOf(min, e)\n    return min\n}\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin("1.4")\npublic actual fun minOf(a: Long, vararg\nother: Long): Long {\n    var min = a\n    for (e in other) min = minOf(min, e)\n    return min\n}\n\n/**\n * Returns\nthe smaller of the given values.\n * \n * If any value is `NaN`, returns `NaN`.\n */\n@SinceKotlin("1.4")\npublic\nactual fun minOf(a: Float, vararg other: Float): Float {\n    var min = a\n    for (e in other) min = minOf(min, e)\n    return min\n}\n\n/**\n * Returns the smaller of the given values.\n * \n * If any value is `NaN`, returns `NaN`.\n */\n@SinceKotlin("1.4")\npublic actual fun minOf(a: Double, vararg other: Double): Double {\n    var min = a\n    for (e in other) min = minOf(min, e)\n    return min\n}\n\n", /*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin\nProgramming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be\nfound in the license/LICENSE.txt file.\n */\n@npackage kotlin\n\nimport\nkotlin.experimental.*\nimport\nkotlin.jvm.*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@JvmInline\npublic value class ULong @kotlin.internal.IntrinsicConstEvaluation @PublishedApi internal\nconstructor(@PublishedApi internal val data: Long) : Comparable<ULong> {\n\n    companion object {\n\n        /**\n         * A constant holding the minimum value an instance of ULong can have.\n         */\n        public const val\nMIN_VALUE: ULong = ULong(0)\n\n        /**\n         * A constant holding the maximum value an instance of

```

```

ULong can have.\n    *^\n    public const val MAX_VALUE: ULong = ULong(-1)\n\n    /**\n    * The number of bytes used to represent an instance of ULong in a binary form.\n    *^\n    public const val SIZE_BYTES: Int = 8\n\n    /**\n    * The number of bits used to represent an instance of ULong in a binary form.\n    *^\n    public const val SIZE_BITS: Int = 64\n    }\n\n    /**\n    * Compares this value with the specified value for order.\n    * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n    * or a positive number if it's greater than other.\n    *^\n    @kotlin.internal.InlineOnly\n    public inline operator fun compareTo(other: UByte): Int = this.compareTo(other.toULong())\n\n    /**\n    * Compares this value with the specified value for order.\n    * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n    * or a positive number if it's greater than other.\n    *^\n    @kotlin.internal.InlineOnly\n    public inline operator fun compareTo(other: UShort): Int = this.compareTo(other.toULong())\n\n    /**\n    * Compares this value with the specified value for order.\n    * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n    * or a positive number if it's greater than other.\n    *^\n    @kotlin.internal.InlineOnly\n    public inline operator fun compareTo(other: UInt): Int = this.compareTo(other.toULong())\n\n    /**\n    * Compares this value with the specified value for order.\n    * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n    * or a positive number if it's greater than other.\n    *^\n    @kotlin.internal.InlineOnly\n    @Suppress("OVERRIDE_BY_INLINE")\n    public override inline operator fun compareTo(other: ULong): Int = ulongCompare(this.data, other.data)\n\n    /** Adds the other value to this value. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun plus(other: UByte): ULong = this.plus(other.toULong())\n\n    /** Adds the other value to this value. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun plus(other: UShort): ULong = this.plus(other.toULong())\n\n    /** Adds the other value to this value. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun plus(other: UInt): ULong = this.plus(other.toULong())\n\n    /** Adds the other value to this value. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun plus(other: ULong): ULong = ULong(this.data.plus(other.data))\n\n    /** Subtracts the other value from this value. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun minus(other: UByte): ULong = this.minus(other.toULong())\n\n    /** Subtracts the other value from this value. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun minus(other: UShort): ULong = this.minus(other.toULong())\n\n    /** Subtracts the other value from this value. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun minus(other: UInt): ULong = this.minus(other.toULong())\n\n    /** Subtracts the other value from this value. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun minus(other: ULong): ULong = ULong(this.data.minus(other.data))\n\n    /** Multiplies this value by the other value. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun times(other: UByte): ULong = this.times(other.toULong())\n\n    /** Multiplies this value by the other value. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun times(other: UShort): ULong = this.times(other.toULong())\n\n    /** Multiplies this value by the other value. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun times(other: UInt): ULong = this.times(other.toULong())\n\n    /** Multiplies this value by the other value. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun times(other: ULong): ULong = ULong(this.data.times(other.data))\n\n    /** Divides this value by the other value, truncating the result to an integer that is closer to zero. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun div(other: UByte): ULong = this.div(other.toULong())\n\n    /** Divides this value by the other value, truncating the result to an integer that is closer to zero. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun div(other: UShort): ULong = this.div(other.toULong())\n\n    /** Divides this value by the other value, truncating the result to an integer that is closer to zero. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun div(other: UInt): ULong = this.div(other.toULong())\n\n    /** Divides this value by the other value, truncating the result to an integer that is closer to zero. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun div(other: ULong): ULong = ulongDivide(this, other)\n\n    /**\n    * Calculates the remainder of truncating division of this value by the other value.\n    * \n    * The result is always less than the divisor.\n    *^\n    @kotlin.internal.InlineOnly\n    public inline operator fun rem(other: UByte): ULong = this.rem(other.toULong())\n\n    /**\n    * Calculates the remainder of truncating division of this value by the other value.\n    * \n    * The result is always less than the divisor.\n    */

```

```

*^/n @kotlin.internal.InlineOnly/n public inline operator fun rem(other: UShort): ULong =
this.rem(other.toULong())/n /**/n * Calculates the remainder of truncating division of this value by the other
value./n * /n * The result is always less than the divisor./n *^/n @kotlin.internal.InlineOnly/n public
inline operator fun rem(other: UInt): ULong = this.rem(other.toULong())/n /**/n * Calculates the remainder of
truncating division of this value by the other value./n * /n * The result is always less than the divisor./n *^/n
@kotlin.internal.InlineOnly/n public inline operator fun rem(other: ULong): ULong = ulongRemainder(this,
other)/n/n /**/n * Divides this value by the other value, flooring the result to an integer that is closer to negative
infinity./n * /n * For unsigned types, the results of flooring division and truncating division are the same./n
*^/n @kotlin.internal.InlineOnly/n public inline fun floorDiv(other: UByte): ULong =
this.floorDiv(other.toULong())/n /**/n * Divides this value by the other value, flooring the result to an integer
that is closer to negative infinity./n * /n * For unsigned types, the results of flooring division and truncating
division are the same./n *^/n @kotlin.internal.InlineOnly/n public inline fun floorDiv(other: UShort): ULong
= this.floorDiv(other.toULong())/n /**/n * Divides this value by the other value, flooring the result to an integer
that is closer to negative infinity./n * /n * For unsigned types, the results of flooring division and truncating
division are the same./n *^/n @kotlin.internal.InlineOnly/n public inline fun floorDiv(other: UInt): ULong =
this.floorDiv(other.toULong())/n /**/n * Divides this value by the other value, flooring the result to an integer
that is closer to negative infinity./n * /n * For unsigned types, the results of flooring division and truncating
division are the same./n *^/n @kotlin.internal.InlineOnly/n public inline fun floorDiv(other: ULong): ULong =
div(other)/n/n /**/n * Calculates the remainder of flooring division of this value by the other value./n * /n *
The result is always less than the divisor./n * /n * For unsigned types, the remainders of flooring division and
truncating division are the same./n *^/n @kotlin.internal.InlineOnly/n public inline fun mod(other: UByte):
UByte = this.mod(other.toULong()).toUByte()/n /**/n * Calculates the remainder of flooring division of this
value by the other value./n * /n * The result is always less than the divisor./n * /n * For unsigned types, the
remainders of flooring division and truncating division are the same./n *^/n @kotlin.internal.InlineOnly/n
public inline fun mod(other: UShort): UShort = this.mod(other.toULong()).toUShort()/n /**/n * Calculates the
remainder of flooring division of this value by the other value./n * /n * The result is always less than the
divisor./n * /n * For unsigned types, the remainders of flooring division and truncating division are the same./n
*^/n @kotlin.internal.InlineOnly/n public inline fun mod(other: UInt): UInt =
this.mod(other.toULong()).toUInt()/n /**/n * Calculates the remainder of flooring division of this value by the
other value./n * /n * The result is always less than the divisor./n * /n * For unsigned types, the remainders
of flooring division and truncating division are the same./n *^/n @kotlin.internal.InlineOnly/n public inline
fun mod(other: ULong): ULong = rem(other)/n/n /**/n * Returns this value incremented by one./n * /n *
@sample samples.misc.Builtins.inc/n *^/n @kotlin.internal.InlineOnly/n public inline operator fun inc():
ULong = ULong(data.inc())/n/n /**/n * Returns this value decremented by one./n * /n * @sample
samples.misc.Builtins.dec/n *^/n @kotlin.internal.InlineOnly/n public inline operator fun dec(): ULong =
ULong(data.dec())/n/n /**/n * Creates a range from this value to the specified [other] value. */n
@kotlin.internal.InlineOnly/n public inline operator fun rangeTo(other: ULong): ULongRange =
ULongRange(this, other)/n/n /**/n * Shifts this value left by the [bitCount] number of bits./n * /n * Note
that only the six lowest-order bits of the [bitCount] are used as the shift distance./n * The shift distance actually
used is therefore always in the range `0..63`./n *^/n @kotlin.internal.InlineOnly/n public inline infix fun
shl(bitCount: Int): ULong = ULong(data shl bitCount)/n/n /**/n * Shifts this value right by the [bitCount]
number of bits, filling the leftmost bits with zeros./n * /n * Note that only the six lowest-order bits of the
[bitCount] are used as the shift distance./n * The shift distance actually used is therefore always in the range
`0..63`./n *^/n @kotlin.internal.InlineOnly/n public inline infix fun shr(bitCount: Int): ULong = ULong(data
ushr bitCount)/n/n /**/n * Performs a bitwise AND operation between the two values. */n
@kotlin.internal.InlineOnly/n public inline infix fun and(other: ULong): ULong = ULong(this.data and
other.data)/n/n /**/n * Performs a bitwise OR operation between the two values. */n @kotlin.internal.InlineOnly/n
public inline infix fun or(other: ULong): ULong = ULong(this.data or other.data)/n/n /**/n * Performs a bitwise XOR

```

```

operation between the two values. */
@kotlin.internal.InlineOnly
public inline infix fun xor(other: ULong):
ULong = ULong(this.data xor other.data) /** Inverts the bits in this value. */
@kotlin.internal.InlineOnly
public inline fun inv(): ULong = ULong(data.inv()) /**
 * Converts this [ULong] value to [Byte].
 * If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value represents
 * the same numerical value as this `ULong`.
 * The resulting `Byte` value is represented by the least significant 8 bits
of this `ULong` value.
 * Note that the resulting `Byte` value may be negative. */
@kotlin.internal.InlineOnly
public inline fun toByte(): Byte = data.toByte() /**
 * Converts this [ULong]
value to [Short].
 * If this value is less than or equals to [Short.MAX_VALUE], the resulting `Short` value
represents
 * the same numerical value as this `ULong`.
 * The resulting `Short` value is represented
by the least significant 16 bits of this `ULong` value.
 * Note that the resulting `Short` value may be negative. */
@kotlin.internal.InlineOnly
public inline fun toShort(): Short = data.toShort() /**
 * Converts this
[ULong] value to [Int].
 * If this value is less than or equals to [Int.MAX_VALUE], the resulting `Int`
value represents
 * the same numerical value as this `ULong`.
 * The resulting `Int` value is
represented by the least significant 32 bits of this `ULong` value.
 * Note that the resulting `Int` value may be
negative. */
@kotlin.internal.InlineOnly
public inline fun toInt(): Int = data.toInt() /**
 *
Converts this [ULong] value to [Long].
 * If this value is less than or equals to [Long.MAX_VALUE], the
resulting `Long` value represents
 * the same numerical value as this `ULong`. Otherwise the result is
negative.
 * The resulting `Long` value has the same binary representation as this `ULong` value. */
@kotlin.internal.InlineOnly
public inline fun toLong(): Long = data.toLong() /**
 * Converts this [ULong]
value to [UByte].
 * If this value is less than or equals to [UByte.MAX_VALUE], the resulting `UByte`
value represents
 * the same numerical value as this `ULong`.
 * The resulting `UByte` value is
represented by the least significant 8 bits of this `ULong` value. */
@kotlin.internal.InlineOnly
public
inline fun toUByte(): UByte = data.toUByte() /**
 * Converts this [ULong] value to [UShort].
 * If
this value is less than or equals to [UShort.MAX_VALUE], the resulting `UShort` value represents
 * the same
numerical value as this `ULong`.
 * The resulting `UShort` value is represented by the least significant 16
bits of this `ULong` value. */
@kotlin.internal.InlineOnly
public inline fun toUShort(): UShort =
data.toUShort() /**
 * Converts this [ULong] value to [UInt].
 * If this value is less than or equals
to [UInt.MAX_VALUE], the resulting `UInt` value represents
 * the same numerical value as this `ULong`.
 * The resulting `UInt` value is represented by the least significant 32 bits of this `ULong` value. */
@kotlin.internal.InlineOnly
public inline fun toUInt(): UInt = data.toUInt() /** Returns this value. */
@kotlin.internal.InlineOnly
public inline fun toULong(): ULong = this.toULong() /**
 * Converts this [ULong]
value to [Float].
 * The resulting value is the closest `Float` to this `ULong` value.
 * In case when this
`ULong` value is exactly between two `Float`s,
 * the one with zero at least significant bit of mantissa is
selected. */
@kotlin.internal.InlineOnly
public inline fun toFloat(): Float = this.toDouble().toFloat() /**
 * Converts this [ULong] value to [Double].
 * The resulting value is the closest `Double` to this
`ULong` value.
 * In case when this `ULong` value is exactly between two `Double`s,
 * the one with zero at
least significant bit of mantissa is selected. */
@kotlin.internal.InlineOnly
public inline fun toDouble():
Double = ulongToDouble(data)
public override fun toString(): String = ulongToString(data) /**
 * Converts this [Byte] value to [ULong].
 * If this value is positive, the resulting `ULong` value represents the
same numerical value as this `Byte`.
 * The least significant 8 bits of the resulting `ULong` value are the same
as the bits of this `Byte` value,
 * whereas the most significant 56 bits are filled with the sign bit of this value. */
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
@kotlin.internal.InlineOnly
public inline fun Byte.toULong(): ULong = ULong(this.toLong()) /**
 * Converts this [Short] value to
[ULong].
 * If this value is positive, the resulting `ULong` value represents the same numerical value as this
`Short`.
 * The least significant 16 bits of the resulting `ULong` value are the same as the bits of this `Short`
value,
 * whereas the most significant 48 bits are filled with the sign bit of this value. */
@SinceKotlin("1.5")
@WasExperimental(ExperimentalUnsignedTypes::class)
@kotlin.internal.InlineOnly
public inline fun Short.toULong(): ULong = ULong(this.toLong()) /**
 * Converts this [Int] value to [ULong]. */

```

*\n * If this value is positive, the resulting `ULong` value represents the same numerical value as this `Int`. \n *\n * The least significant 32 bits of the resulting `ULong` value are the same as the bits of this `Int` value, \n * whereas the most significant 32 bits are filled with the sign bit of this value. \n

```

*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Int.toULong(): ULong = ULong(this.toLong())\n*\n * Converts this [Long] value to [ULong]. \n
*\n * If this value is positive, the resulting `ULong` value represents the same numerical value as this `Long`. \n *\n * The resulting `ULong` value has the same binary representation as this `Long` value. \n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Long.toULong(): ULong = ULong(this)\n*\n * Converts this [Float] value to [ULong]. \n *\n * The fractional part, if any, is rounded down towards zero. \n * Returns zero if this `Float` value is negative or `NaN`, [ULong.MAX_VALUE] if it's bigger than `ULong.MAX_VALUE`. \n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Float.toULong(): ULong = doubleToULong(this.toDouble())\n*\n * Converts this [Double] value to [ULong]. \n *\n * The fractional part, if any, is rounded down towards zero. \n * Returns zero if this `Double` value is negative or `NaN`, [ULong.MAX_VALUE] if it's bigger than `ULong.MAX_VALUE`. \n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Double.toULong(): ULong = doubleToULong(this)\n", "\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file. \n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("CollectionsKt")\n\npackage kotlin.collections\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.*\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n*\n * Returns 1st *element* from the list. \n * \n * Throws an [IndexOutOfBoundsException] if the size of this list is less than 1. \n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> List<T>.component1(): T {\n    return get(0)\n}\n*\n * Returns 2nd *element* from the list. \n * \n * Throws an [IndexOutOfBoundsException] if the size of this list is less than 2. \n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> List<T>.component2(): T {\n    return get(1)\n}\n*\n * Returns 3rd *element* from the list. \n * \n * Throws an [IndexOutOfBoundsException] if the size of this list is less than 3. \n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> List<T>.component3(): T {\n    return get(2)\n}\n*\n * Returns 4th *element* from the list. \n * \n * Throws an [IndexOutOfBoundsException] if the size of this list is less than 4. \n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> List<T>.component4(): T {\n    return get(3)\n}\n*\n * Returns 5th *element* from the list. \n * \n * Throws an [IndexOutOfBoundsException] if the size of this list is less than 5. \n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> List<T>.component5(): T {\n    return get(4)\n}\n*\n * Returns `true` if [element] is found in the collection. \n
*\npublic operator fun <@kotlin.internal.OnlyInputTypes T> Iterable<T>.contains(element: T): Boolean {\n    if (this is Collection)\n        return contains(element)\n    return indexOf(element) >= 0\n}\n*\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this collection. \n * \n * @sample samples.collections.Collections.Elements.elementAt\n*\npublic fun <T> Iterable<T>.elementAt(index: Int): T {\n    if (this is List)\n        return get(index)\n    return elementAtOrElse(index) {\n        throw IndexOutOfBoundsException("Collection doesn't contain element at index $index.")\n    }\n}\n*\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this list. \n * \n * @sample samples.collections.Collections.Elements.elementAt\n*\n@kotlin.internal.InlineOnly\npublic inline fun <T> List<T>.elementAt(index: Int): T {\n    return get(index)\n}\n*\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this collection. \n * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n*\npublic fun <T> Iterable<T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {\n    if (this is List)\n        return

```

```

this.getOrElse(index, defaultValue)\n if (index < 0)\n     return defaultValue(index)\n val iterator = iterator()\n var count = 0\n while (iterator.hasNext()) {\n     val element = iterator.next()\n     if (index == count++)\n         return element\n } \n return defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the\n result of calling the [defaultValue] function if the [index] is out of bounds of this list.\n * \n * @sample\n samples.collections.Collections.Elements.elementAtOrElse\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun\n<T> List<T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {\n    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the\n [index] is out of bounds of this collection.\n * \n * @sample\n samples.collections.Collections.Elements.elementAtOrNull\n */\n\npublic fun <T>\nIterable<T>.elementAtOrNull(index: Int): T? {\n    if (this is List)\n        return this.elementAtOrNull(index)\n    if (index < 0)\n        return null\n    val iterator = iterator()\n    var count = 0\n    while (iterator.hasNext()) {\n        val element = iterator.next()\n        if (index == count++)\n            return element\n    } \n    return null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this list.\n * \n * @sample\n samples.collections.Collections.Elements.elementAtOrNull\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun\n<T> List<T>.elementAtOrNull(index: Int): T? {\n    return this.elementAtOrNull(index)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample\n samples.collections.Collections.Elements.find\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun <T>\nIterable<T>.find(predicate: (T) -> Boolean): T? {\n    return firstOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample\n samples.collections.Collections.Elements.find\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun <T>\nIterable<T>.findLast(predicate: (T) -> Boolean): T? {\n    return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample\n samples.collections.Collections.Elements.find\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun <T>\nList<T>.findLast(predicate: (T) -> Boolean): T? {\n    return lastOrNull(predicate)\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n\npublic fun <T>\nIterable<T>.first(): T {\n    when (this) {\n        is List -> return this.first()\n        else -> {\n            val iterator = iterator()\n            if (!iterator.hasNext())\n                throw NoSuchElementException("Collection is empty.")\n            return iterator.next()\n        }\n    }\n}\n\n/**\n * Returns the first element.\n * \n * @throws\n NoSuchElementException if the list is empty.\n */\n\npublic fun <T> List<T>.first(): T {\n    if (isEmpty())\n        throw NoSuchElementException("List is empty.")\n    return this[0]\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * \n * @throws [NoSuchElementException] if no such element is found.\n */\n\npublic inline fun <T> Iterable<T>.first(predicate: (T) -> Boolean): T {\n    for (element in this) if (predicate(element))\n        return element\n    throw NoSuchElementException("Collection contains no element matching the\n predicate.")\n}\n\n/**\n * Returns the first non-null value produced by [transform] function being applied to\n elements of this collection in iteration order,\n * or throws [NoSuchElementException] if no non-null value was produced.\n * \n * @sample\n samples.collections.Collections.Transformations.firstNotNullOf\n */\n\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any>\nIterable<T>.firstNotNullOf(transform: (T) -> R?): R {\n    return firstNotNullOfOrNull(transform) ?: throw\n NoSuchElementException("No element of the collection was transformed to a non-null value.")\n}\n\n/**\n * Returns the first non-null value produced by [transform] function being applied to elements of this collection in\n iteration order,\n * or `null` if no non-null value was produced.\n * \n * @sample\n samples.collections.Collections.Transformations.firstNotNullOf\n */\n\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any>\nIterable<T>.firstNotNullOfOrNull(transform: (T) -> R?): R? {\n    for (element in this) {\n        val result = transform(element)\n        if (result != null) {\n            return result\n        }\n    }\n    return null\n}\n\n/**\n * Returns the first element, or `null` if the collection is empty.\n */\n\npublic fun <T> Iterable<T>.firstOrNull(): T? {\n    when (this) {\n        is List -> {\n            if (isEmpty())\n                return null\n            else\n                return this[0]\n        }\n        else -> {\n            val iterator = iterator()\n            if (!iterator.hasNext())\n                return null\n        }\n    }\n}

```



```

return iterator.next()\n    }\n }\n}\n\n/**\n * Returns the first element, or `null` if the list is empty.\n */\npublic\nfun <T> List<T>.firstOrNull(): T? {\n    return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element\n matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun <T>\nIterable<T>.firstOrNull(predicate: (T) -> Boolean): T? {\n    for (element in this) if (predicate(element)) return\n element\n    return null\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the\n [defaultValue] function if the [index] is out of bounds of this list.\n */\n@kotlin.internal.InlineOnly\npublic inline\nfun <T> List<T>.getOrNull(index: Int, default: (Int) -> T): T {\n    return if (index >= 0 && index <=\n lastIndex) get(index) else default(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the\n [index] is out of bounds of this list.\n */\n * @sample samples.collections.Collections.Elements.getOrNull\n */\npublic fun <T> List<T>.getOrNull(index: Int): T? {\n    return if (index >= 0 && index <= lastIndex) get(index)\n else null\n}\n\n/**\n * Returns first index of [element], or -1 if the collection does not contain element.\n */\npublic fun <@kotlin.internal.OnlyInputTypes T> Iterable<T>.indexOf(element: T): Int {\n    if (this is List) return\n this.indexOf(element)\n    var index = 0\n    for (item in this) {\n        checkIndexOverflow(index)\n        if (element\n == item)\n            return index\n        index++\n    }\n    return -1\n}\n\n/**\n * Returns first index of [element], or -1\n if the list does not contain element.\n */\n@Suppress("EXTENSION_SHADOWED_BY_MEMBER") // false\nwarning, extension takes precedence in some cases\npublic fun <@kotlin.internal.OnlyInputTypes T>\nList<T>.indexOf(element: T): Int {\n    return indexOf(element)\n}\n\n/**\n * Returns index of the first element\n matching the given [predicate], or -1 if the collection does not contain such element.\n */\npublic inline fun <T>\nIterable<T>.indexOfFirst(predicate: (T) -> Boolean): Int {\n    var index = 0\n    for (item in this) {\n        checkIndexOverflow(index)\n        if (predicate(item))\n            return index\n        index++\n    }\n    return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the list does not contain\n such element.\n */\npublic inline fun <T> List<T>.indexOfFirst(predicate: (T) -> Boolean): Int {\n    var index = 0\n    for (item in this) {\n        if (predicate(item))\n            return index\n        index++\n    }\n    return -1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the collection does not contain such\n element.\n */\npublic inline fun <T> Iterable<T>.indexOfLast(predicate: (T) -> Boolean): Int {\n    var lastIndex = -1\n    var index = 0\n    for (item in this) {\n        checkIndexOverflow(index)\n        if (predicate(item))\n            lastIndex = index\n        index++\n    }\n    return lastIndex\n}\n\n/**\n * Returns index of the last element matching\n the given [predicate], or -1 if the list does not contain such element.\n */\npublic inline fun <T>\nList<T>.indexOfLast(predicate: (T) -> Boolean): Int {\n    val iterator = this.listIterator(size)\n    while\n (iterator.hasPrevious()) {\n        if (predicate(iterator.previous()))\n            return iterator.nextIndex()\n    }\n    return -1\n}\n\n/**\n * Returns the last element.\n */\n * @throws NoSuchElementException if the collection\n is empty.\n */\n * @sample samples.collections.Collections.Elements.last\n */\npublic fun <T> Iterable<T>.last(): T\n {\n    when (this) {\n        is List -> return this.last()\n        else -> {\n            val iterator = iterator()\n            if\n (!iterator.hasNext())\n                throw NoSuchElementException("Collection is empty.")\n            var last =\n iterator.next()\n            while (iterator.hasNext())\n                last = iterator.next()\n            return last\n        }\n    }\n}\n\n/**\n * Returns the last element.\n */\n * @throws NoSuchElementException if the list is empty.\n */\n * @sample samples.collections.Collections.Elements.last\n */\npublic fun <T> List<T>.last(): T {\n    if (isEmpty())\n        throw NoSuchElementException("List is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last\n element matching the given [predicate].\n */\n * @throws NoSuchElementException if no such element is found.\n */\n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun <T>\nIterable<T>.last(predicate: (T) -> Boolean): T {\n    var last: T? = null\n    var found = false\n    for (element in this)\n {\n        if (predicate(element)) {\n            last = element\n            found = true\n        }\n    }\n    if (!found) throw\n NoSuchElementException("Collection contains no element matching the predicate.")\n    @Suppress("UNCHECKED_CAST")\n    return last as T\n}\n\n/**\n * Returns the last element matching the\n given [predicate].\n */\n * @throws NoSuchElementException if no such element is found.\n */\n * @sample\n samples.collections.Collections.Elements.last\n */\npublic inline fun <T> List<T>.last(predicate: (T) -> Boolean): T\n {\n    val iterator = this.listIterator(size)\n    while (iterator.hasPrevious()) {\n        val element = iterator.previous()\n        if (predicate(element)) return element\n    }\n    throw NoSuchElementException("List contains no element

```

```

matching the predicate.}n}n/n/**n * Returns last index of [element], or -1 if the collection does not contain
element.}n */npublic fun <@kotlin.internal.OnlyInputTypes T> Iterable<T>.lastIndexOf(element: T): Int {n if
(this is List) return this.lastIndexOf(element)n var lastIndex = -1n var index = 0n for (item in this) {n
checkIndexOverflow(index)n if (element == item)n lastIndex = indexn index++n }n return
lastIndexn}n/n/**n * Returns last index of [element], or -1 if the list does not contain element.}n
*/n@Suppress("EXTENSION_SHADOWED_BY_MEMBER") // false warning, extension takes precedence in
some casesnpublic fun <@kotlin.internal.OnlyInputTypes T> List<T>.lastIndexOf(element: T): Int {n return
lastIndexOf(element)n}n/n/**n * Returns the last element, or `null` if the collection is empty.}n */n * @sample
samples.collections.Collections.Elements.lastn */npublic fun <T> Iterable<T>.lastOrNull(): T? {n when (this)
{n is List -> return if (isEmpty()) null else this[size - 1]n else -> {n val iterator = iterator()n
if (!iterator.hasNext())n return nulln var last = iterator.next()n while (iterator.hasNext())n
last = iterator.next()n return lastn }n }n}n/n/**n * Returns the last element, or `null` if the
list is empty.}n */n * @sample samples.collections.Collections.Elements.lastn */npublic fun <T>
List<T>.lastOrNull(): T? {n return if (isEmpty()) null else this[size - 1]n}n/n/**n * Returns the last element
matching the given [predicate], or `null` if no such element was found.}n */n * @sample
samples.collections.Collections.Elements.lastn */npublic inline fun <T> Iterable<T>.lastOrNull(predicate: (T) ->
Boolean): T? {n var last: T? = nulln for (element in this) {n if (predicate(element)) {n last =
elementn }n }n return lastn}n/n/**n * Returns the last element matching the given [predicate], or `null`
if no such element was found.}n */n * @sample samples.collections.Collections.Elements.lastn */npublic inline
fun <T> List<T>.lastOrNull(predicate: (T) -> Boolean): T? {n val iterator = this.listIterator(size)n while
(iterator.hasPrevious()) {n val element = iterator.previous()n if (predicate(element)) return elementn }n
return nulln}n/n/**n * Returns a random element from this collection.}n */n * @throws
NoSuchElementException if this collection is empty.}n
*/n@SinceKotlin("1.3")n@kotlin.internal.InlineOnlynpublic inline fun <T> Collection<T>.random(): T {n
return random(Random)n}n/n/**n * Returns a random element from this collection using the specified source of
randomness.}n */n * @throws NoSuchElementException if this collection is empty.}n
*/n@SinceKotlin("1.3")npublic fun <T> Collection<T>.random(random: Random): T {n if (isEmpty())n
throw NoSuchElementException("Collection is empty.")}n return elementAt(random.nextInt(size))n}n/n/**n *
Returns a random element from this collection, or `null` if this collection is empty.}n
*/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)n@kotlin.internal.InlineOnlynpublic
inline fun <T> Collection<T>.randomOrNull(): T? {n return randomOrNull(Random)n}n/n/**n * Returns a
random element from this collection using the specified source of randomness, or `null` if this collection is empty.}n
*/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)npublic fun <T>
Collection<T>.randomOrNull(random: Random): T? {n if (isEmpty())n return nulln return
elementAt(random.nextInt(size))n}n/n/**n * Returns the single element, or throws an exception if the collection is
empty or has more than one element.}n */npublic fun <T> Iterable<T>.single(): T {n when (this) {n is List -
-> return this.single()n else -> {n val iterator = iterator()n if (!iterator.hasNext())n
throw NoSuchElementException("Collection is empty.")}n val single = iterator.next()n if
(iterator.hasNext())n throw IllegalArgumentException("Collection has more than one element.")}n
return singlen }n }n}n/n/**n * Returns the single element, or throws an exception if the list is empty or
has more than one element.}n */npublic fun <T> List<T>.single(): T {n return when (size) {n 0 -> throw
NoSuchElementException("List is empty.")}n 1 -> this[0]n else -> throw
IllegalArgumentException("List has more than one element.")}n }n}n/n/**n * Returns the single element
matching the given [predicate], or throws exception if there is no or more than one matching element.}n */npublic
inline fun <T> Iterable<T>.single(predicate: (T) -> Boolean): T {n var single: T? = nulln var found = falsen
for (element in this) {n if (predicate(element)) {n if (found) throw
IllegalArgumentException("Collection contains more than one matching element.")}n single = elementn
found = true\n }n }n if (!found) throw NoSuchElementException("Collection contains no element

```

```

matching the predicate.\n @Suppress("UNCHECKED_CAST")\n return single as T\n\n**\n * Returns
single element, or `null` if the collection is empty or has more than one element.\n */\npublic fun <T>
Iterable<T>.singleOrNull(): T? {\n    when (this) {\n        is List -> return if (size == 1) this[0] else null\n        else ->
{\n            val iterator = iterator()\n            if (!iterator.hasNext())\n                return null\n            val single =
iterator.next()\n            if (iterator.hasNext())\n                return null\n            return single\n        }\n    }\n\n**\n * Returns single element, or `null` if the list is empty or has more than one element.\n */\npublic fun <T>
List<T>.singleOrNull(): T? {\n    return if (size == 1) this[0] else null\n\n**\n * Returns the single element
matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\npublic
inline fun <T> Iterable<T>.singleOrNull(predicate: (T) -> Boolean): T? {\n    var single: T? = null\n    var found =
false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single =
element\n            found = true\n        }\n    }\n    if (!found) return null\n    return single\n\n**\n * Returns a list
containing all elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n *
\n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun <T> Iterable<T>.drop(n: Int):
List<T> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    if (n == 0) return toList()\n    val list: ArrayList<T>\n    if (this is Collection<*>) {\n        val resultSize = size - n\n        if (resultSize <= 0)\n            return emptyList()\n        if (resultSize == 1)\n            return listOf(last())\n        list = ArrayList<T>(resultSize)\n        if (this is List<T>) {\n            if (this is RandomAccess) {\n                for (index in n until size)\n                    list.add(this[index])\n            } else {\n                for (item in listIterator(n))\n                    list.add(item)\n            }\n            return list\n        }\n    } else {\n        list = ArrayList<T>()\n        var count = 0\n        for (item in this) {\n            if (count >= n) list.add(item) else ++count\n        }\n        return list.optimizeReadOnlyList()\n\n**\n * Returns a list
containing all elements except last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n *
\n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun <T> List<T>.dropLast(n: Int):
List<T> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    return take((size -
n).coerceAtLeast(0))\n\n**\n * Returns a list containing all elements except last elements that satisfy the given
[predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun <T>
List<T>.dropLastWhile(predicate: (T) -> Boolean): List<T> {\n    if (!isEmpty()) {\n        val iterator =
listIterator(size)\n        while (iterator.hasPrevious()) {\n            if (!predicate(iterator.previous())) {\n                return
take(iterator.nextIndex() + 1)\n            }\n        }\n    }\n    return emptyList()\n\n**\n * Returns a list containing
all elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic inline fun <T> Iterable<T>.dropWhile(predicate:
(T) -> Boolean): List<T> {\n    var yielding = false\n    val list = ArrayList<T>()\n    for (item in this)\n        if
(yielding)\n            list.add(item)\n        else if (!predicate(item)) {\n            list.add(item)\n            yielding = true\n        }\n    return list\n\n**\n * Returns a list containing only elements matching the given [predicate].\n * \n *
@sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun <T> Iterable<T>.filter(predicate: (T)
-> Boolean): List<T> {\n    return filterTo(ArrayList<T>(), predicate)\n\n**\n * Returns a list containing only
elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the
element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun <T>
Iterable<T>.filterIndexed(predicate: (index: Int, T) -> Boolean): List<T> {\n    return
filterIndexedTo(ArrayList<T>(), predicate)\n\n**\n * Appends all elements matching the given [predicate] to
the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n *
and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <T, C : MutableCollection<in T>>
Iterable<T>.filterIndexedTo(destination: C, predicate: (index: Int, T) -> Boolean): C {\n    forEachIndexed { index,
element ->\n        if (predicate(index, element)) destination.add(element)\n    }\n    return destination\n\n**\n *
Returns a list containing all elements that are instances of specified type parameter R.\n * \n * @sample
samples.collections.Collections.Filtering.filterIsInstance\n */\npublic inline fun <reified R>
Iterable<*>.filterIsInstance(): List<@kotlin.internal.NoInfer R> {\n    return

```

```

filterIsInstanceTo(Iterable<R>())\n\n/**\n * Appends all elements that are instances of specified type
parameter R to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterIsInstanceTo\n *\npublic inline fun <reified R, C :
MutableCollection<in R>> Iterable<*>.filterIsInstanceTo(destination: C): C {\n for (element in this) if (element is
R) destination.add(element)\n return destination}\n\n/**\n * Returns a list containing all elements not matching
the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n *\npublic inline fun <T>
Iterable<T>.filterNot(predicate: (T) -> Boolean): List<T> {\n return filterNotTo(Iterable<T>()),
predicate)\n}\n\n/**\n * Returns a list containing all elements that are not `null`.\n * \n * @sample
samples.collections.Collections.Filtering.filterNotNull\n *\npublic fun <T : Any> Iterable<T?>.filterNotNull():
List<T> {\n return filterNotNullTo(Iterable<T?>())}\n\n/**\n * Appends all elements that are not `null` to the
given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterNotNullTo\n *\npublic fun <C
: MutableCollection<in T>, T : Any> Iterable<T?>.filterNotNullTo(destination: C): C {\n for (element in this) if
(element != null) destination.add(element)\n return destination}\n\n/**\n * Appends all elements not matching
the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n
*\npublic inline fun <T, C : MutableCollection<in T>> Iterable<T>.filterTo(destination: C, predicate: (T) ->
Boolean): C {\n for (element in this) if (!predicate(element)) destination.add(element)\n return
destination}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * \n *
@sample samples.collections.Collections.Filtering.filterTo\n *\npublic inline fun <T, C : MutableCollection<in
T>> Iterable<T>.filterTo(destination: C, predicate: (T) -> Boolean): C {\n for (element in this) if
(predicate(element)) destination.add(element)\n return destination}\n}\n\n/**\n * Returns a list containing elements
at indices in the specified [indices] range.\n *\npublic fun <T> List<T>.slice(indices: IntRange): List<T> {\n if
(indices.isEmpty()) return listOf()\n return this.subList(indices.start, indices.endInclusive + 1).toList()\n}\n\n/**\n
* Returns a list containing elements at specified [indices].\n *\npublic fun <T> List<T>.slice(indices:
Iterable<Int>): List<T> {\n val size = indices.collectionSizeOrDefault(10)\n if (size == 0) return emptyList()\n
val list = ArrayList<T>(size)\n for (index in indices) {\n list.add(get(index))\n }\n return list}\n\n/**\n
* Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n *
@sample samples.collections.Collections.Transformations.take\n *\npublic fun <T> Iterable<T>.take(n: Int):
List<T> {\n require(n >= 0) { "Requested element count $n is less than zero." }\n if (n == 0) return
emptyList()\n if (this is Collection<T>) {\n if (n >= size) return toList()\n if (n == 1) return
listOf(first())\n }\n var count = 0\n val list = ArrayList<T>(n)\n for (item in this) {\n list.add(item)\n
if (++count == n)\n break\n }\n return list.optimizeReadOnlyList()\n}\n\n/**\n * Returns a list containing
last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n *\npublic fun <T> List<T>.takeLast(n: Int): List<T> {\n
require(n >= 0) { "Requested element count $n is less than zero." }\n if (n == 0) return emptyList()\n val size =
size\n if (n >= size) return toList()\n if (n == 1) return listOf(last())\n val list = ArrayList<T>(n)\n if (this is
RandomAccess) {\n for (index in size - n until size)\n list.add(this[index])\n } else {\n for (item in
listIterator(size - n))\n list.add(item)\n }\n return list}\n\n/**\n * Returns a list containing last elements
satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n *\npublic
inline fun <T> List<T>.takeLastWhile(predicate: (T) -> Boolean): List<T> {\n if (isEmpty())\n return
emptyList()\n val iterator = listIterator(size)\n while (iterator.hasPrevious()) {\n if
(!predicate(iterator.previous())) {\n iterator.next()\n val expectedSize = size - iterator.nextIndex()\n
if (expectedSize == 0) return emptyList()\n return ArrayList<T>(expectedSize).apply {\n while
(iterator.hasNext())\n add(iterator.next())\n }\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing
first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *\npublic inline fun <T> Iterable<T>.takeWhile(predicate:
(T) -> Boolean): List<T> {\n val list = ArrayList<T>()\n for (item in this) {\n if (!predicate(item))\n
break\n list.add(item)\n }\n return list}\n\n/**\n * Reverses elements in the list in-place.\n *\npublic
expect fun <T> MutableList<T>.reverse(): Unit\n\n/**\n * Returns a list with elements in reversed order.\n

```

```

*public fun <T> Iterable<T>.reversed(): List<T> {
    if (this is Collection && size <= 1) return toList()
    val list = toMutableList()
    list.reverse()
    return list
}

* Randomly shuffles elements in this list in-place using the specified [random] instance as the source of randomness.
* See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm

*@SinceKotlin("1.3")
*public fun <T> MutableList<T>.shuffle(random: Random): Unit {
    for (i in lastIndex downTo 1) {
        val j = random.nextInt(i + 1)
        this[j] = this.set(i, this[j])
    }
}

* Sorts elements in the list in-place according to natural sort order of the value returned by specified [selector] function.
* The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.

*public inline fun <T, R : Comparable<R>> MutableList<T>.sortBy(crossinline selector: (T) -> R?): Unit {
    if (size > 1)
        sortWith(compareBy(selector))
}

* Sorts elements in the list in-place descending according to natural sort order of the value returned by specified [selector] function.
* The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.

*public inline fun <T, R : Comparable<R>> MutableList<T>.sortByDescending(crossinline selector: (T) -> R?): Unit {
    if (size > 1)
        sortWith(compareByDescending(selector))
}

* Sorts elements in the list in-place descending according to their natural sort order.
* The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.

*public fun <T : Comparable<T>> MutableList<T>.sortDescending(): Unit {
    sortWith(reverseOrder())
}

* Returns a list of all elements sorted according to their natural sort order.
* The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.

*public fun <T : Comparable<T>> Iterable<T>.sorted(): List<T> {
    if (this is Collection) {
        if (size <= 1)
            return this.toList()
        @Suppress("UNCHECKED_CAST")
        return (toArray<Comparable<T>>() as Array<T>).apply { sort() }.asList()
    }
    return toMutableList().apply { sort() }
}

* Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.
* The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.

*sample samples.collections.Collections.Sorting.sortedBy
*public inline fun <T, R : Comparable<R>> Iterable<T>.sortedBy(crossinline selector: (T) -> R?): List<T> {
    return sortedWith(compareBy(selector))
}

* Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.
* The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.

*public inline fun <T, R : Comparable<R>> Iterable<T>.sortedByDescending(crossinline selector: (T) -> R?): List<T> {
    return sortedWith(compareByDescending(selector))
}

* Returns a list of all elements sorted descending according to their natural sort order.
* The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.

*public fun <T : Comparable<T>> Iterable<T>.sortedDescending(): List<T> {
    return sortedWith(reverseOrder())
}

* Returns a list of all elements sorted according to the specified [comparator].
* The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.

*public fun <T> Iterable<T>.sortedWith(comparator: Comparator<in T>): List<T> {
    if (this is Collection) {
        if (size <= 1)
            return this.toList()
        @Suppress("UNCHECKED_CAST")
        return (toArray<Any?>() as Array<T>).apply { sortWith(comparator) }.asList()
    }
    return toMutableList().apply { sortWith(comparator) }
}

* Returns an array of Boolean containing all of the elements of this collection.

*public fun Collection<Boolean>.toBooleanArray(): BooleanArray {
    val result = BooleanArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

* Returns an array of Byte containing all of the elements of this collection.

*public fun Collection<Byte>.toByteArray(): ByteArray {
    val result = ByteArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

* Returns an array of Char containing all of the elements of this collection.

*public fun Collection<Char>.toCharArray(): CharArray {
    val result = CharArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

* Returns an array of Double containing all of the elements of this collection.

*public fun Collection<Double>.toDoubleArray(): DoubleArray {
    val result = DoubleArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

* Returns an array of Float containing

```

```

all of the elements of this collection.\n */\npublic fun Collection<Float>.toFloatArray(): FloatArray {\n    val result = FloatArray(size)\n    var index = 0\n    for (element in this)\n        result[index++] = element\n    return result\n}\n\n/**\n * Returns an array of Int containing all of the elements of this collection.\n */\npublic fun Collection<Int>.toIntArray(): IntArray {\n    val result = IntArray(size)\n    var index = 0\n    for (element in this)\n        result[index++] = element\n    return result\n}\n\n/**\n * Returns an array of Long containing all of the elements of this collection.\n */\npublic fun Collection<Long>.toLongArray(): LongArray {\n    val result = LongArray(size)\n    var index = 0\n    for (element in this)\n        result[index++] = element\n    return result\n}\n\n/**\n * Returns an array of Short containing all of the elements of this collection.\n */\npublic fun Collection<Short>.toShortArray(): ShortArray {\n    val result = ShortArray(size)\n    var index = 0\n    for (element in this)\n        result[index++] = element\n    return result\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to elements of the given collection.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original collection.\n * \n * @sample samples.collections.Collections.Transformations.associate\n */\npublic inline fun <T, K, V> Iterable<T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> {\n    val capacity = mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)\n    return associateTo(LinkedHashMap<K, V>(capacity), transform)\n}\n\n/**\n * Returns a [Map] containing the elements from the given collection indexed by the key\n * returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original collection.\n * \n * @sample samples.collections.Collections.Transformations.associateBy\n */\npublic inline fun <T, K> Iterable<T>.associateBy(keySelector: (T) -> K): Map<K, T> {\n    val capacity = mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, T>(capacity), keySelector)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given collection.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original collection.\n * \n * @sample samples.collections.Collections.Transformations.associateByWithValueTransform\n */\npublic inline fun <T, K, V> Iterable<T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> {\n    val capacity = mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given collection\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Collections.Transformations.associateByTo\n */\npublic inline fun <T, K, M : MutableMap<in K, in T>> Iterable<T>.associateByTo(destination: M, keySelector: (T) -> K): M {\n    for (element in this) {\n        destination.put(keySelector(element), element)\n    }\n    return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * value is provided by the [valueTransform] function applied to elements of the given collection.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Collections.Transformations.associateByToWithValueTransform\n */\npublic inline fun <T, K, V, M : MutableMap<in K, in V>> Iterable<T>.associateByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {\n    for (element in this) {\n        destination.put(keySelector(element), valueTransform(element))\n    }\n    return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each element of the given collection.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample samples.collections.Collections.Transformations.associateTo\n */\npublic inline fun <T, K, V, M : MutableMap<in K, in V>> Iterable<T>.associateTo(destination: M, transform: (T) -> Pair<K, V>): M {\n    for (element in this) {\n        destination += transform(element)\n    }\n    return destination\n}\n\n/**\n * Returns a [Map] where keys are

```

elements from the given collection and values are produced by the [valueSelector] function applied to each element. If any two elements are equal, the last one gets added to the map. The returned map preserves the entry iteration order of the original collection.

@sample
samples.collections.Collections.Transformations.associateWith

```

*^@SinceKotlin("1.3")
public inline fun <K, V> Iterable<K>.associateWith(valueSelector: (K) -> V): Map<K, V> {
    val result = LinkedHashMap<K, V>(mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16))
    return associateWithTo(result, valueSelector)
}

```

Populates and returns the [destination] mutable map with key-value pairs for each element of the given collection, where key is the element itself and value is provided by the [valueSelector] function applied to that key. If any two elements are equal, the last one overwrites the former value in the map.

@sample samples.collections.Collections.Transformations.associateWithTo

```

*^@SinceKotlin("1.3")
public inline fun <K, V, M : MutableMap<in K, in V>>
Iterable<K>.associateWithTo(destination: M, valueSelector: (K) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

```

Appends all elements to the given [destination] collection.

```

*^@SinceKotlin("1.3")
public fun <T, C : MutableCollection<in T>>
Iterable<T>.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

```

Returns a new [HashSet] of all elements.

```

*^@SinceKotlin("1.3")
public fun <T> Iterable<T>.toHashSet():
HashSet<T> {
    return toCollection(HashSet<T>(mapCapacity(collectionSizeOrDefault(12))))
}

```

Returns a [List] containing all elements.

```

*^@SinceKotlin("1.3")
public fun <T> Iterable<T>.toList(): List<T> {
    if (this is Collection) {
        return when (size) {
            0 -> emptyList()
            1 -> listOf(if (this is List) get(0) else iterator().next())
            else -> this.toMutableList()
        }
    }
    return this.toMutableList().optimizeReadOnlyList()
}

```

Returns a new [MutableList] filled with all elements of this collection.

```

*^@SinceKotlin("1.3")
public fun <T> Iterable<T>.toMutableList(): MutableList<T> {
    if (this is Collection<T>) {
        return this.toMutableList()
    }
    return toCollection(ArrayList<T>())
}

```

Returns a new [MutableList] filled with all elements of this collection.

```

*^@SinceKotlin("1.3")
public fun <T> Collection<T>.toMutableList(): MutableList<T> {
    return ArrayList(this)
}

```

Returns a [Set] of all elements. The returned set preserves the element iteration order of the original collection.

```

*^@SinceKotlin("1.3")
public fun <T> Iterable<T>.toSet(): Set<T> {
    if (this is Collection) {
        return when (size) {
            0 -> emptySet()
            1 -> setOf(if (this is List) this[0] else iterator().next())
            else -> toCollection(LinkedHashSet<T>(mapCapacity(size)))
        }
    }
    return toCollection(LinkedHashSet<T>()).optimizeReadOnlySet()
}

```

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original collection.

@sample
samples.collections.Collections.Transformations.flatMap

```

*^@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.jvm.JvmName("flatMapSequence")
public inline fun <T, R>
Iterable<T>.flatMap(transform: (T) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

```

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original collection.

@sample
samples.collections.Collections.Transformations.flatMap

```

*^@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterable")
@kotlin.internal.InlineOnly
public inline fun <T, R> Iterable<T>.flatMapIndexed(transform: (index: Int, T) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

```

Returns a single list of all elements yielded from results of [transform] function being invoked on each element and its index in the original collection.

@sample
samples.collections.Collections.Transformations.flatMapIndexed

```

*^@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexed")
@kotlin.internal.InlineOnly
public inline fun <T, R> Iterable<T>.flatMapIndexed(transform: (index: Int, T) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

```

```

ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapIndexedSequence")\n@kotlin.internal.InlineOnly\npubli
c inline fun <T, R> Iterable<T>.flatMapIndexed(transform: (index: Int, T) -> Sequence<R>): List<R> {\n  return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original collection, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <T, R, C : MutableCollection<in R>> Iterable<T>.flatMapIndexedTo(destination: C, transform: (index:
Int, T) -> Iterable<R>): C {\n  var index = 0\n  for (element in this) {\n    val list =
transform(checkIndexOverflow(index++), element)\n    destination.addAll(list)\n  }\n  return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element\n * and its index in the original collection, to the given [destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapIndexedSequenceTo")\n@kotlin.internal.InlineOnly\npu
blic inline fun <T, R, C : MutableCollection<in R>> Iterable<T>.flatMapIndexedTo(destination: C, transform:
(index: Int, T) -> Sequence<R>): C {\n  var index = 0\n  for (element in this) {\n    val list =
transform(checkIndexOverflow(index++), element)\n    destination.addAll(list)\n  }\n  return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original collection, to the given [destination].\n * \npublic inline fun <T, R, C : MutableCollection<in
R>> Iterable<T>.flatMapTo(destination: C, transform: (T) -> Iterable<R>): C {\n  for (element in this) {\n    val
list = transform(element)\n    destination.addAll(list)\n  }\n  return destination\n}\n\n/**\n * Appends all
elements yielded from results of [transform] function being invoked on each element of original collection, to the
given [destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapSequenceTo")\npublic inline fun <T, R, C :
MutableCollection<in R>> Iterable<T>.flatMapTo(destination: C, transform: (T) -> Sequence<R>): C {\n  for
(element in this) {\n    val list = transform(element)\n    destination.addAll(list)\n  }\n  return
destination\n}\n\n/**\n * Groups elements of the original collection by the key returned by the given [keySelector]
function\n * applied to each element and returns a map where each group key is associated with a list of
corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the
original collection.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n * \npublic inline
fun <T, K> Iterable<T>.groupBy(keySelector: (T) -> K): Map<K, List<T>> {\n  return
groupByTo(LinkedHashMap<K, MutableList<T>>(), keySelector)\n}\n\n/**\n * Groups values returned by the
[valueTransform] function applied to each element of the original collection\n * by the key returned by the given
[keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of
corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the
original collection.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n * \npublic
inline fun <T, K, V> Iterable<T>.groupBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K,
List<V>> {\n  return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector,
valueTransform)\n}\n\n/**\n * Groups elements of the original collection by the key returned by the given
[keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with
a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n * \npublic inline fun <T, K, M : MutableMap<in K,
MutableList<T>>> Iterable<T>.groupByTo(destination: M, keySelector: (T) -> K): M {\n  for (element in this) {\n
    val key = keySelector(element)\n    val list = destination.getOrPut(key) { ArrayList<T>() }\n
list.add(element)\n  }\n  return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function
applied to each element of the original collection\n * by the key returned by the given [keySelector] function applied
to the element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n

```



```

 * \n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n * \n public inline fun <T, K, V, M :
MutableMap<in K, MutableList<V>>> Iterable<T>.groupByTo(destination: M, keySelector: (T) -> K,
valueTransform: (T) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return
destination}\n}\n\n/**\n * Creates a [Grouping] source from a collection to be used later with one of group-and-fold
operations\n * using the specified [keySelector] function to extract a key from each element.\n * \n * @sample
samples.collections.Grouping.groupingByEachCount\n * \n @SinceKotlin("1.1")\n public inline fun <T, K>
Iterable<T>.groupingBy(crossinline keySelector: (T) -> K): Grouping<T, K> {\n return object : Grouping<T, K>
{\n override fun sourceIterator(): Iterator<T> = this@groupingBy.iterator()\n override fun keyOf(element:
T): K = keySelector(element)\n }\n}\n}\n\n/**\n * Returns a list containing the results of applying the given
[transform] function\n * to each element in the original collection.\n * \n * @sample
samples.collections.Collections.Transformations.map\n * \n public inline fun <T, R> Iterable<T>.map(transform:
(T) -> R): List<R> {\n return mapTo(ArrayList<R>(collectionSizeOrDefault(10)), transform)\n}\n}\n\n/**\n *
Returns a list containing the results of applying the given [transform] function\n * to each element and its index in
the original collection.\n * @param [transform] function that takes the index of an element and the element itself\n *
and returns the result of the transform applied to the element.\n * \n public inline fun <T, R>
Iterable<T>.mapIndexed(transform: (index: Int, T) -> R): List<R> {\n return
mapIndexedTo(ArrayList<R>(collectionSizeOrDefault(10)), transform)\n}\n}\n\n/**\n * Returns a list containing only
the non-null results of applying the given [transform] function\n * to each element and its index in the original
collection.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns
the result of the transform applied to the element.\n * \n public inline fun <T, R : Any>
Iterable<T>.mapIndexedNotNull(transform: (index: Int, T) -> R?): List<R> {\n return
mapIndexedNotNullTo(ArrayList<R>(), transform)\n}\n}\n\n/**\n * Applies the given [transform] function to each
element and its index in the original collection\n * and appends only the non-null results to the given [destination].\n *
@param [transform] function that takes the index of an element and the element itself\n * and returns the result of
the transform applied to the element.\n * \n public inline fun <T, R : Any, C : MutableCollection<in R>>
Iterable<T>.mapIndexedNotNullTo(destination: C, transform: (index: Int, T) -> R?): C {\n forEachIndexed {
index, element -> transform(index, element)?.let { destination.add(it) } }\n return destination}\n}\n}\n\n/**\n *
Applies the given [transform] function to each element and its index in the original collection\n * and appends the
results to the given [destination].\n * @param [transform] function that takes the index of an element and the
element itself\n * and returns the result of the transform applied to the element.\n * \n public inline fun <T, R, C :
MutableCollection<in R>> Iterable<T>.mapIndexedTo(destination: C, transform: (index: Int, T) -> R): C {\n var
index = 0\n for (item in this)\n destination.add(transform(checkIndexOverflow(index++), item))\n return
destination}\n}\n}\n\n/**\n * Returns a list containing only the non-null results of applying the given [transform]
function\n * to each element in the original collection.\n * \n * @sample
samples.collections.Collections.Transformations.mapNotNull\n * \n public inline fun <T, R : Any>
Iterable<T>.mapNotNull(transform: (T) -> R?): List<R> {\n return mapNotNullTo(ArrayList<R>(),
transform)\n}\n}\n\n/**\n * Applies the given [transform] function to each element in the original collection\n * and
appends only the non-null results to the given [destination].\n * \n public inline fun <T, R : Any, C :
MutableCollection<in R>> Iterable<T>.mapNotNullTo(destination: C, transform: (T) -> R?): C {\n forEach {
element -> transform(element)?.let { destination.add(it) } }\n return destination}\n}\n}\n\n/**\n * Applies the given
[transform] function to each element of the original collection\n * and appends the results to the given
[destination].\n * \n public inline fun <T, R, C : MutableCollection<in R>> Iterable<T>.mapTo(destination: C,
transform: (T) -> R): C {\n for (item in this)\n destination.add(transform(item))\n return
destination}\n}\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original collection\n * into an
[IndexValue] containing the index of that element and the element itself.\n * \n public fun <T>
Iterable<T>.withIndex(): Iterable<IndexedValue<T>> {\n return IndexingIterable { iterator() }\n}\n}\n\n/**\n *

```

Returns a list containing only distinct elements from the given collection.

- Among equal elements of the given collection, only the first one will be present in the resulting list.
- The elements in the resulting list are in the same order as they were in the source collection.

@sample

```
samples.collections.Collections.Transformations.distinctAndDistinctBy
```

`public fun <T> Iterable<T>.distinct(): List<T>` {
 return this.toMutableSet().toList()
}

Returns a list containing only elements from the given collection having distinct keys returned by the given [selector] function.

- Among elements of the given collection with equal keys, only the first one will be present in the resulting list.
- The elements in the resulting list are in the same order as they were in the source collection.

@sample

```
samples.collections.Collections.Transformations.distinctAndDistinctBy
```

`public inline fun <T, K> Iterable<T>.distinctBy(selector: (T) -> K): List<T>` {
 val set = HashSet<K>()
 val list = ArrayList<T>()
 for (e in this) {
 val key = selector(e)
 if (set.add(key))
 list.add(e)
 }
 return list
}

Returns a set containing all elements that are contained by both this collection and the specified collection.

The returned set preserves the element iteration order of the original collection.

To get a set containing all elements that are contained at least in one of these collections use [union].

```
public infix fun <T> Iterable<T>.intersect(other: Iterable<T>): Set<T>
```

{
 val set = this.toMutableSet()
 set.retainAll(other)
 return set
}

Returns a set containing all elements that are contained by this collection and not contained by the specified collection.

The returned set preserves the element iteration order of the original collection.

```
public infix fun <T> Iterable<T>.subtract(other: Iterable<T>): Set<T>
```

{
 val set = this.toMutableSet()
 set.removeAll(other)
 return set
}

Returns a new [MutableSet] containing all distinct elements from the given collection.

The returned set preserves the element iteration order of the original collection.

```
public fun <T> Iterable<T>.toMutableSet(): MutableSet<T>
```

{
 return when (this) {
 is Collection<T> -> LinkedHashSet(this)
 else -> toCollection(LinkedHashSet<T>())
 }
}

Returns a set containing all distinct elements from both collections.

The returned set preserves the element iteration order of the original collection.

Those elements of the [other] collection that are unique are iterated in the end in the order of the [other] collection.

To get a set containing all elements that are contained in both collections use [intersect].

```
public infix fun <T> Iterable<T>.union(other: Iterable<T>): Set<T>
```

{
 val set = this.toMutableSet()
 set.addAll(other)
 return set
}

Returns `true` if all elements match the given [predicate].

@sample

```
samples.collections.Collections.Aggregates.all
```

`public inline fun <T> Iterable<T>.all(predicate: (T) -> Boolean): Boolean` {
 if (this is Collection && isEmpty()) return true
 for (element in this) if (!predicate(element)) return false
 return true
}

Returns `true` if collection has at least one element.

@sample

```
samples.collections.Collections.Aggregates.any
```

`public fun <T> Iterable<T>.any(): Boolean` {
 if (this is Collection) return !isEmpty()
 return iterator().hasNext()
}

Returns `true` if at least one element matches the given [predicate].

@sample

```
samples.collections.Collections.Aggregates.anyWithPredicate
```

`public inline fun <T> Iterable<T>.any(predicate: (T) -> Boolean): Boolean` {
 if (this is Collection && isEmpty()) return false
 for (element in this) if (predicate(element)) return true
 return false
}

Returns the number of elements in this collection.

```
public fun <T> Iterable<T>.count(): Int
```

{
 if (this is Collection) return size
 var count = 0
 for (element in this) checkCountOverflow(++count)
 return count
}

Returns the number of elements in this collection.

```
@kotlin.internal.InlineOnly public inline fun <T> Collection<T>.count(): Int
```

{
 return size
}

Returns the number of elements matching the given [predicate].

```
public inline fun <T> Iterable<T>.count(predicate: (T) -> Boolean): Int
```

{
 if (this is Collection && isEmpty()) return 0
 var count = 0
 for (element in this) if (predicate(element)) checkCountOverflow(++count)
 return count
}

Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

Returns the specified [initial] value if the collection is empty.

@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

```
public inline fun <T, R> Iterable<T>.fold(initial: R, operation: (acc: R, T) -> R): R
```

{
 var accumulator = initial
 for (element in this) accumulator = operation(accumulator, element)
 return accumulator
}

Accumulates value starting with [initial] value and applying [operation] from left to

```

right\n * to current accumulator value and each element with its index in the original collection.\n * \n * Returns the
specified [initial] value if the collection is empty.\n * \n * @param [operation] function that takes the index of an
element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n
*/\npublic inline fun <T, R> Iterable<T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {\n  var
index = 0\n  var accumulator = initial\n  for (element in this) accumulator =
operation(checkIndexOverflow(index++), accumulator, element)\n  return accumulator\n}\n\n/**\n * Accumulates
value starting with [initial] value and applying [operation] from right to left\n * to each element and current
accumulator value.\n * \n * Returns the specified [initial] value if the list is empty.\n * \n * @param [operation]
function that takes an element and current accumulator value, and calculates the next accumulator value.\n
*/\npublic inline fun <T, R> List<T>.foldRight(initial: R, operation: (T, acc: R) -> R): R {\n  var accumulator =
initial\n  if (!isEmpty()) {\n    val iterator = listIterator(size)\n    while (iterator.hasPrevious()) {\n
accumulator = operation(iterator.previous(), accumulator)\n    }\n  }\n  return accumulator\n}\n\n/**\n *
Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element with
its index in the original list and current accumulator value.\n * \n * Returns the specified [initial] value if the list is
empty.\n * \n * @param [operation] function that takes the index of an element, the element itself\n * and current
accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <T, R>
List<T>.foldRightIndexed(initial: R, operation: (index: Int, T, acc: R) -> R): R {\n  var accumulator = initial\n  if
(!isEmpty()) {\n    val iterator = listIterator(size)\n    while (iterator.hasPrevious()) {\n      val index =
iterator.previousIndex()\n      accumulator = operation(index, iterator.previous(), accumulator)\n    }\n  }\n  return accumulator\n}\n\n/**\n * Performs the given [action] on each element.\n
*/\n@kotlin.internal.HidesMembers\npublic inline fun <T> Iterable<T>.forEach(action: (T) -> Unit): Unit {\n  for
(element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential
index with the element.\n * \n * @param [action] function that takes the index of an element and the element itself\n *
and performs the action on the element.\n */\npublic inline fun <T> Iterable<T>.forEachIndexed(action: (index: Int,
T) -> Unit): Unit {\n  var index = 0\n  for (item in this) action(checkIndexOverflow(index++), item)\n}\n\n/**\n *
Returns the largest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws
NoSuchElementException if the collection is empty.\n
*/\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrNull")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
Iterable<Double>.max(): Double {\n  val iterator = iterator()\n  if (!iterator.hasNext()) throw
NoSuchElementException()\n  var max = iterator.next()\n  while (iterator.hasNext()) {\n    val e =
iterator.next()\n    max = maxOf(max, e)\n  }\n  return max\n}\n\n/**\n * Returns the largest element.\n * \n *
If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the collection is empty.\n
*/\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrNull")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
Iterable<Float>.max(): Float {\n  val iterator = iterator()\n  if (!iterator.hasNext()) throw
NoSuchElementException()\n  var max = iterator.next()\n  while (iterator.hasNext()) {\n    val e =
iterator.next()\n    max = maxOf(max, e)\n  }\n  return max\n}\n\n/**\n * Returns the largest element.\n * \n *
@throws NoSuchElementException if the collection is empty.\n
*/\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrNull")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
<T : Comparable<T>> Iterable<T>.max(): T {\n  val iterator = iterator()\n  if
(!iterator.hasNext()) throw NoSuchElementException()\n  var max = iterator.next()\n  while (iterator.hasNext())
{\n    val e = iterator.next()\n    if (max < e) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first
element yielding the largest value of the given function.\n * \n * @throws NoSuchElementException if the
collection is empty.\n * \n * @sample samples.collections.Collections.Aggregates.maxBy\n
*/\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrNull")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun
<T, R : Comparable<R>> Iterable<T>.maxBy(selector: (T) -> R): T {\n  val iterator =
iterator()\n  if (!iterator.hasNext()) throw NoSuchElementException()\n  var maxElem = iterator.next()\n  if
(!iterator.hasNext()) return maxElem\n  var maxValue = selector(maxElem)\n  do {\n    val e = iterator.next()\n
val v = selector(e)\n    if (maxValue < v) {\n      maxElem = e\n      maxValue = v\n    }\n  } while

```

```

(iterator.hasNext())\n    return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the
given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.maxByOrNull\n * \n * @SinceKotlin("1.4")\n\npublic inline fun <T, R :
Comparable<R>> Iterable<T>.maxByOrNull(selector: (T) -> R): T? {\n    val iterator = iterator()\n    if
(!iterator.hasNext()) return null\n    var maxElem = iterator.next()\n    if (!iterator.hasNext()) return maxElem\n    var
maxValue = selector(maxElem)\n    do {\n        val e = iterator.next()\n        val v = selector(e)\n        if (maxValue <
v) {\n            maxElem = e\n            maxValue = v\n        }\n    } while (iterator.hasNext())\n    return
maxElem\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the collection.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result
is `NaN`.\n * \n * @throws NoSuchElementException if the collection is empty.\n
*\n * \n * @SinceKotlin("1.4")\n\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n\n@OverloadResolution
ByLambdaReturnType\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <T> Iterable<T>.maxOf(selector: (T) ->
Double): Double {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var
maxValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n
maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all
values produced by [selector] function\n * applied to each element in the collection.\n * \n * If any of values
produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException
if the collection is empty.\n
*\n * \n * @SinceKotlin("1.4")\n\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n\n@OverloadResolution
ByLambdaReturnType\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <T> Iterable<T>.maxOf(selector: (T) ->
Float): Float {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var
maxValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n
maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all
values produced by [selector] function\n * applied to each element in the collection.\n * \n * @throws
NoSuchElementException if the collection is empty.\n
*\n * \n * @SinceKotlin("1.4")\n\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n\n@OverloadResolution
ByLambdaReturnType\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <T, R : Comparable<R>>
Iterable<T>.maxOf(selector: (T) -> R): R {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw
NoSuchElementException()\n    var maxValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v
= selector(iterator.next())\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the collection or `null` if there are no elements.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n
*\n * \n * @SinceKotlin("1.4")\n\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n\n@OverloadResolution
ByLambdaReturnType\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <T> Iterable<T>.maxOfOrNull(selector: (T)
-> Double): Double? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var maxValue =
selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        maxValue =
maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the collection or `null` if there are no elements.\n * \n * If
any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n * \n * @SinceKotlin("1.4")\n\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n\n@OverloadResolution
ByLambdaReturnType\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <T> Iterable<T>.maxOfOrNull(selector: (T)
-> Float): Float? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var maxValue =
selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        maxValue =
maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the collection or `null` if there are no elements.\n
*\n * \n * @SinceKotlin("1.4")\n\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n\n@OverloadResolution
ByLambdaReturnType\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <T, R : Comparable<R>>

```

```

Iterable<T>.maxOrNull(selector: (T) -> R): R? {\n  val iterator = iterator()\n  if (!iterator.hasNext()) return\n  null\n  var maxValue = selector(iterator.next())\n  while (iterator.hasNext()) {\n    val v =\n    selector(iterator.next())\n    if (maxValue < v) {\n      maxValue = v\n    }\n  }\n  return\n  maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values\n * produced by [selector] function applied to each element in the collection.\n * \n * @throws\n * NoSuchElementException if the collection is empty.\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>\n\nIterable<T>.maxOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n  val iterator = iterator()\n  if\n  (!iterator.hasNext()) throw NoSuchElementException()\n  var maxValue = selector(iterator.next())\n  while\n  (iterator.hasNext()) {\n    val v = selector(iterator.next())\n    if (comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value according to the provided\n * [comparator]\n * among all values produced by [selector] function applied to each element in the collection or `null`\n * if there are no elements.\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>\n\nIterable<T>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n  val iterator =\n  iterator()\n  if (!iterator.hasNext()) return null\n  var maxValue = selector(iterator.next())\n  while\n  (iterator.hasNext()) {\n    val v = selector(iterator.next())\n    if (comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest element or `null` if there are no\n * elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n\n*\n@SinceKotlin("1.4")\npublic fun\n\nIterable<Double>.maxOrNull(): Double? {\n  val iterator = iterator()\n  if (!iterator.hasNext()) return null\n  var\n  max = iterator.next()\n  while (iterator.hasNext()) {\n    val e = iterator.next()\n    max = maxOf(max, e)\n  }\n  return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n * \n * If any of\n * elements is `NaN` returns `NaN`.\n\n*\n@SinceKotlin("1.4")\npublic fun Iterable<Float>.maxOrNull(): Float? {\n  val\n  iterator = iterator()\n  if (!iterator.hasNext()) return null\n  var max = iterator.next()\n  while\n  (iterator.hasNext()) {\n    val e = iterator.next()\n    max = maxOf(max, e)\n  }\n  return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n\n*\n@SinceKotlin("1.4")\npublic fun <T :\nComparable<T>> Iterable<T>.maxOrNull(): T? {\n  val iterator = iterator()\n  if (!iterator.hasNext()) return null\n  var\n  max = iterator.next()\n  while (iterator.hasNext()) {\n    val e = iterator.next()\n    if (max < e) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided\n * [comparator].\n * \n * @throws NoSuchElementException if the collection is empty.\n\n*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow")\n@Suppress("CONFLICTING_OVER\nLOADS")\npublic fun <T> Iterable<T>.maxWith(comparator: Comparator<in T>): T {\n  val iterator =\n  iterator()\n  if (!iterator.hasNext()) throw NoSuchElementException()\n  var max = iterator.next()\n  while\n  (iterator.hasNext()) {\n    val e = iterator.next()\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator]\n * or `null` if there are no elements.\n\n*\n@SinceKotlin("1.4")\npublic fun <T>\n\nIterable<T>.maxWithOrNull(comparator: Comparator<in T>): T? {\n  val iterator = iterator()\n  if\n  (!iterator.hasNext()) return null\n  var max = iterator.next()\n  while (iterator.hasNext()) {\n    val e =\n    iterator.next()\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the\n * smallest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if\n * the collection is empty.\n\n*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOA\nDS")\npublic fun Iterable<Double>.min(): Double {\n  val iterator = iterator()\n  if (!iterator.hasNext()) throw\n  NoSuchElementException()\n  var min = iterator.next()\n  while (iterator.hasNext()) {\n    val e =\n    iterator.next()\n    min = minOf(min, e)\n  }\n  return min\n}\n\n/**\n * Returns the smallest element.\n * \n * If\n * any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the collection is empty.\n\n*\n
```

```

*\/n@SinceKotlin("1.7")n@kotlin.jvm.JvmName("minOrThrow")n@Suppress("CONFLICTING_OVERLOA
DS")npublic fun Iterable<Float>.min(): Float {n  val iterator = iterator()n  if (!iterator.hasNext()) throw
NoSuchElementException()n  var min = iterator.next()n  while (iterator.hasNext()) {n    val e =
iterator.next()n    min = minOf(min, e)n  }n  return min\n}n/n/**n * Returns the smallest element.n * n *
@throws NoSuchElementException if the collection is empty.n
*\/n@SinceKotlin("1.7")n@kotlin.jvm.JvmName("minOrThrow")n@Suppress("CONFLICTING_OVERLOA
DS")npublic fun <T : Comparable<T>> Iterable<T>.min(): T {n  val iterator = iterator()n  if
(!iterator.hasNext()) throw NoSuchElementException()n  var min = iterator.next()n  while (iterator.hasNext())
{n    val e = iterator.next()n    if (min > e) min = e\n  }n  return min\n}n/n/**n * Returns the first element
yielding the smallest value of the given function.n * n * @throws NoSuchElementException if the collection is
empty.n * n * @sample samples.collections.Collections.Aggregates.minBy\n
*\/n@SinceKotlin("1.7")n@kotlin.jvm.JvmName("minByOrThrow")n@Suppress("CONFLICTING_OVERLO
ADS")npublic inline fun <T, R : Comparable<R>> Iterable<T>.minBy(selector: (T) -> R): T {n  val iterator =
iterator()n  if (!iterator.hasNext()) throw NoSuchElementException()n  var minElem = iterator.next()n  if
(!iterator.hasNext()) return minElem\n  var minValue = selector(minElem)n  do {n    val e = iterator.next()n
    val v = selector(e)n    if (minValue > v) {n      minElem = e\n      minValue = v\n    }n  } while
(iterator.hasNext())n  return minElem\n}n/n/**n * Returns the first element yielding the smallest value of the
given function or `null` if there are no elements.n * n * @sample
samples.collections.Collections.Aggregates.minByOrNull\n
*\/n@SinceKotlin("1.4")npublic inline fun <T, R :
Comparable<R>> Iterable<T>.minByOrNull(selector: (T) -> R): T? {n  val iterator = iterator()n  if
(!iterator.hasNext()) return null\n  var minElem = iterator.next()n  if (!iterator.hasNext()) return minElem\n  var
minValue = selector(minElem)n  do {n    val e = iterator.next()n    val v = selector(e)n    if (minValue >
v) {n      minElem = e\n      minValue = v\n    }n  } while (iterator.hasNext())n  return
minElem\n}n/n/**n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the collection.n * n * If any of values produced by [selector] function is `NaN`, the returned result
is `NaN`.n * n * @throws NoSuchElementException if the collection is empty.n
*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun <T> Iterable<T>.minOf(selector: (T) ->
Double): Double {n  val iterator = iterator()n  if (!iterator.hasNext()) throw NoSuchElementException()n  var
minValue = selector(iterator.next())n  while (iterator.hasNext()) {n    val v = selector(iterator.next())n
minValue = minOf(minValue, v)n  }n  return minValue\n}n/n/**n * Returns the smallest value among all
values produced by [selector] function\n * applied to each element in the collection.n * n * If any of values
produced by [selector] function is `NaN`, the returned result is `NaN`.n * n * @throws NoSuchElementException
if the collection is empty.n
*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun <T> Iterable<T>.minOf(selector: (T) ->
Float): Float {n  val iterator = iterator()n  if (!iterator.hasNext()) throw NoSuchElementException()n  var
minValue = selector(iterator.next())n  while (iterator.hasNext()) {n    val v = selector(iterator.next())n
minValue = minOf(minValue, v)n  }n  return minValue\n}n/n/**n * Returns the smallest value among all
values produced by [selector] function\n * applied to each element in the collection.n * n * @throws
NoSuchElementException if the collection is empty.n
*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun <T, R : Comparable<R>>
Iterable<T>.minOf(selector: (T) -> R): R {n  val iterator = iterator()n  if (!iterator.hasNext()) throw
NoSuchElementException()n  var minValue = selector(iterator.next())n  while (iterator.hasNext()) {n    val v
= selector(iterator.next())n    if (minValue > v) {n      minValue = v\n    }n  }n  return
minValue\n}n/n/**n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the collection or `null` if there are no elements.n * n * If any of values produced by [selector]

```

function is `NaN`, the returned result is `NaN`.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOfOrNull(selector: (T)\n-> Double): Double? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var minValue =\n    selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        minValue =\n        minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced\n * by [selector] function\n * applied to each element in the collection or `null` if there are no elements.\n * \n * If any\n * of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOfOrNull(selector: (T)\n-> Float): Float? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var minValue =\n    selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        minValue =\n        minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced\n * by [selector] function\n * applied to each element in the collection or `null` if there are no elements.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>\nIterable<T>.minOfOrNull(selector: (T) -> R): R? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return\n    null\n    var minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v =\n        selector(iterator.next())\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return\n    minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values\n * produced by [selector] function applied to each element in the collection.\n * \n * @throws\n * NoSuchElementException if the collection is empty.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Iterable<T>.minOfWith(comparator:\nComparator<in R>, selector: (T) -> R): R {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw\n    NoSuchElementException()\n    var minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v\n        = selector(iterator.next())\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all\n * values produced by [selector] function applied to each element in the collection or `null` if there are no elements.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>\nIterable<T>.minOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n    val iterator =\n    iterator()\n    if (!iterator.hasNext()) return null\n    var minValue = selector(iterator.next())\n    while\n    (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest element or `null` if there are\n * no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n\n@SinceKotlin("1.4")\npublic fun
```

```
Iterable<Double>.minOrNull(): Double? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var\n    min = iterator.next()\n    while (iterator.hasNext()) {\n        val e = iterator.next()\n        min = minOf(min, e)\n    }\n    return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n * \n * If any of elements\n * is `NaN` returns `NaN`.\n */
```

```
@SinceKotlin("1.4")\npublic fun Iterable<Float>.minOrNull(): Float? {\n    val\n    iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var min = iterator.next()\n    while (iterator.hasNext())\n    {\n        val e = iterator.next()\n        min = minOf(min, e)\n    }\n    return min\n}\n\n/**\n * Returns the smallest\n * element or `null` if there are no elements.\n */\n\n@SinceKotlin("1.4")\npublic fun <T : Comparable<T>>
```

```
Iterable<T>.minOrNull(): T? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var min =\n    iterator.next()\n    while (iterator.hasNext()) {\n        val e = iterator.next()\n        if (min > e) min = e\n    }\n    return\n    min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * \n * @throws NoSuchElementException if the collection is empty.\n */
```

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVER
```

```

LOADS`)
public fun <T> Iterable<T>.minWith(comparator: Comparator<in T>): T {
    val iterator = iterator()
    if (!iterator.hasNext()) throw NoSuchElementException()
    var min = iterator.next()
    while (iterator.hasNext()) {
        val e = iterator.next()
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}
// Returns the first element having the smallest value according to the provided [comparator] or
// `null` if there are no elements.
@SinceKotlin("1.4")
public fun <T>
Iterable<T>.minOrNull(comparator: Comparator<in T>): T? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var min = iterator.next()
    while (iterator.hasNext()) {
        val e =
iterator.next()
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}
// Returns `true` if
the collection has no elements.
@sample samples.collections.Collections.Aggregates.none
public fun
<T> Iterable<T>.none(): Boolean {
    if (this is Collection) return isEmpty()
    return
!iterator().hasNext()
}
// Returns `true` if no elements match the given [predicate].
@sample
samples.collections.Collections.Aggregates.noneWithPredicate
public inline fun <T>
Iterable<T>.none(predicate: (T) -> Boolean): Boolean {
    if (this is Collection && isEmpty()) return true
    for (element in this) if (predicate(element)) return false
    return true
}
// Performs the given [action] on each
element and returns the collection itself afterwards.
@SinceKotlin("1.1")
public inline fun <T, C :
Iterable<T>> C.onEach(action: (T) -> Unit): C {
    return apply { for (element in this) action(element)
}
}
// Performs the given [action] on each element, providing sequential index with the element,
// and
returns the collection itself afterwards.
@param [action] function that takes the index of an element and the
element itself
// and performs the action on the element.
@SinceKotlin("1.4")
public inline fun <T, C :
Iterable<T>> C.onEachIndexed(action: (index: Int, T) -> Unit): C {
    return apply { forEachIndexed(action)
}
}
// Accumulates value starting with the first element and applying [operation] from left to right
// to
current accumulator value and each element.
// Throws an exception if this collection is empty. If the
collection can be empty in an expected way,
// please use [reduceOrNull] instead. It returns `null` when its receiver
is empty.
@param [operation] function that takes current accumulator value and an element,
// and
calculates the next accumulator value.
@sample samples.collections.Collections.Aggregates.reduce
public inline fun <S, T : S> Iterable<T>.reduce(operation: (acc: S, T) -> S): S {
    val iterator = this.iterator()
    if (!iterator.hasNext()) throw UnsupportedOperationException("Empty collection can't be reduced.")
    var
accumulator: S = iterator.next()
    while (iterator.hasNext()) {
        accumulator = operation(accumulator,
iterator.next())
    }
    return accumulator
}
// Accumulates value starting with the first element and
applying [operation] from left to right
// to current accumulator value and each element with its index in the
original collection.
// Throws an exception if this collection is empty. If the collection can be empty in an
expected way,
// please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.
@param [operation] function that takes the index of an element, current accumulator value and the element itself,
// and
calculates the next accumulator value.
@sample samples.collections.Collections.Aggregates.reduce
public inline fun <S, T : S> Iterable<T>.reduceIndexed(operation: (index: Int, acc: S, T) -> S): S {
    val
iterator = this.iterator()
    if (!iterator.hasNext()) throw UnsupportedOperationException("Empty collection can't
be reduced.")
    var index = 1
    var accumulator: S = iterator.next()
    while (iterator.hasNext()) {
        accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())
    }
    return
accumulator
}
// Accumulates value starting with the first element and applying [operation] from left to
right
// to current accumulator value and each element with its index in the original collection.
// Returns
`null` if the collection is empty.
@param [operation] function that takes the index of an element, current
accumulator value and the element itself,
// and
calculates the next accumulator value.
@sample
samples.collections.Collections.Aggregates.reduceOrNull
@SinceKotlin("1.4")
public inline fun <S, T : S>
Iterable<T>.reduceIndexedOrNull(operation: (index: Int, acc: S, T) -> S): S? {
    val iterator = this.iterator()
    if (!iterator.hasNext()) return null
    var index = 1
    var accumulator: S = iterator.next()
    while (iterator.hasNext()) {
        accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())
    }
    return
accumulator
}
// Accumulates value starting with the first element and applying [operation]
from left to right
// to current accumulator value and each element.
// Returns `null` if the collection is

```



```

empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n * \n * @SinceKotlin("1.4")\n * @WasExperimental(ExperimentalStdlibApi::class)\n * public inline fun <S, T : S> Iterable<T>.reduceOrNull(operation: (acc: S, T) -> S): S? {\n *     val iterator = this.iterator()\n *     if (!iterator.hasNext()) return null\n *     var accumulator: S = iterator.next()\n *     while (iterator.hasNext()) {\n *         accumulator = operation(accumulator, iterator.next())\n *     }\n *     return accumulator\n * }\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Throws an exception if this list is empty. If the list can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n * \n * public inline fun <S, T : S> List<T>.reduceRight(operation: (T, acc: S) -> S): S {\n *     val iterator = listIterator(size)\n *     if (!iterator.hasPrevious()) throw UnsupportedOperationException("Empty list can't be reduced.")\n *     var accumulator: S = iterator.previous()\n *     while (iterator.hasPrevious()) {\n *         accumulator = operation(iterator.previous(), accumulator)\n *     }\n *     return accumulator\n * }\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original list and current accumulator value.\n * \n * Throws an exception if this list is empty. If the list can be empty in an expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n * \n * public inline fun <S, T : S> List<T>.reduceRightIndexed(operation: (index: Int, T, acc: S) -> S): S {\n *     val iterator = listIterator(size)\n *     if (!iterator.hasPrevious()) throw UnsupportedOperationException("Empty list can't be reduced.")\n *     var accumulator: S = iterator.previous()\n *     while (iterator.hasPrevious()) {\n *         val index = iterator.previousIndex()\n *         accumulator = operation(index, iterator.previous(), accumulator)\n *     }\n *     return accumulator\n * }\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original list and current accumulator value.\n * \n * Returns `null` if the list is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n * \n * @SinceKotlin("1.4")\n * public inline fun <S, T : S> List<T>.reduceRightIndexedOrNull(operation: (index: Int, T, acc: S) -> S): S? {\n *     val iterator = listIterator(size)\n *     if (!iterator.hasPrevious()) return null\n *     var accumulator: S = iterator.previous()\n *     while (iterator.hasPrevious()) {\n *         val index = iterator.previousIndex()\n *         accumulator = operation(index, iterator.previous(), accumulator)\n *     }\n *     return accumulator\n * }\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns `null` if the list is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n * \n * @SinceKotlin("1.4")\n * @WasExperimental(ExperimentalStdlibApi::class)\n * public inline fun <S, T : S> List<T>.reduceRightOrNull(operation: (T, acc: S) -> S): S? {\n *     val iterator = listIterator(size)\n *     if (!iterator.hasPrevious()) return null\n *     var accumulator: S = iterator.previous()\n *     while (iterator.hasPrevious()) {\n *         accumulator = operation(iterator.previous(), accumulator)\n *     }\n *     return accumulator\n * }\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n * \n * @SinceKotlin("1.4")\n * public inline fun <T, R>

```

```

Iterable<T>.runningFold(initial: R, operation: (acc: R, T) -> R): List<R> {
    val estimatedSize = collectionSizeOrDefault(9)
    if (estimatedSize == 0) return listOf(initial)
    val result = ArrayList<R>(estimatedSize + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original collection and current accumulator value that starts with [initial] value.

Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
@sample
samples.collections.Collections.Aggregates.runningFold

^@SinceKotlin("1.4")
public inline fun <T, R>
Iterable<T>.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {
    val estimatedSize = collectionSizeOrDefault(9)
    if (estimatedSize == 0) return listOf(initial)
    val result = ArrayList<R>(estimatedSize + 1).apply { add(initial) }
    var index = 0
    var accumulator = initial
    for (element in this) {
        accumulator = operation(index++, accumulator, element)
        result.add(accumulator)
    }
    return result
}

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this collection.

Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

@param [operation] function that takes current accumulator value and the element, and calculates the next accumulator value.
@sample
samples.collections.Collections.Aggregates.runningReduce

^@SinceKotlin("1.4")
^@WasExperimental(ExperimentalStdlibApi::class)
public inline fun <S, T : S>
Iterable<T>.runningReduce(operation: (acc: S, T) -> S): List<S> {
    val iterator = this.iterator()
    if (!iterator.hasNext()) return emptyList()
    var accumulator: S = iterator.next()
    val result = ArrayList<S>(collectionSizeOrDefault(10)).apply { add(accumulator) }
    while (iterator.hasNext()) {
        accumulator = operation(accumulator, iterator.next())
        result.add(accumulator)
    }
    return result
}

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original collection and current accumulator value that starts with the first element of this collection.

Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
@sample
samples.collections.Collections.Aggregates.runningReduce

^@SinceKotlin("1.4")
public inline fun <S, T : S>
Iterable<T>.runningReduceIndexed(operation: (index: Int, acc: S, T) -> S): List<S> {
    val iterator = this.iterator()
    if (!iterator.hasNext()) return emptyList()
    var accumulator: S = iterator.next()
    val result = ArrayList<S>(collectionSizeOrDefault(10)).apply {
        add(accumulator)
    }
    var index = 1
    while (iterator.hasNext()) {
        accumulator = operation(index++, accumulator, iterator.next())
        result.add(accumulator)
    }
    return result
}

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value.

Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
@sample
samples.collections.Collections.Aggregates.scan

^@SinceKotlin("1.4")
^@WasExperimental(ExperimentalStdlibApi::class)
public inline fun <T, R>
Iterable<T>.scan(initial: R, operation: (acc: R, T) -> R): List<R> {
    return runningFold(initial, operation)
}

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original collection and current accumulator value that starts with [initial] value.

Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator

```

```

value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <T, R>
Iterable<T>.scanIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {\n return
runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the collection.\n *\n@Deprecated("Use sumOf instead.\",
ReplaceWith("this.sumOf(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun <T>
Iterable<T>.sumBy(selector: (T) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum +=
selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the collection.\n *\n@Deprecated("Use sumOf instead.\",
ReplaceWith("this.sumOf(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun <T>
Iterable<T>.sumByDouble(selector: (T) -> Double): Double {\n var sum: Double = 0.0\n for (element in this)
{\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the collection.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
<T> Iterable<T>.sumOf(selector: (T) -> Double): Double {\n var sum: Double = 0.toDouble()\n for (element in
this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the collection.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Iterable<T>.sumOf(selector: (T) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this) {\n sum +=
selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the collection.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun
<T> Iterable<T>.sumOf(selector: (T) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this) {\n
sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the collection.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.sumOf(selector: (T) -> UInt): UInt {\n
var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the
collection.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.sumOf(selector: (T) -> ULong): ULong
{\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return
sum\n}\n\n/**\n * Returns an original collection containing all the non-`null` elements, throwing an
[IllegalArgumentException] if there are any `null` elements.\n *\npublic fun <T : Any>
Iterable<T?>.requireNotNulls(): Iterable<T> {\n for (element in this) {\n if (element == null) {\n throw
IllegalArgumentException("null element found in $this.\")\n }\n }\n
@Suppress("UNCHECKED_CAST")\n return this as Iterable<T>\n}\n\n/**\n * Returns an original collection
containing all the non-`null` elements, throwing an [IllegalArgumentException] if there are any `null` elements.\n
*\npublic fun <T : Any> List<T?>.requireNotNulls(): List<T> {\n for (element in this) {\n if (element ==
null) {\n throw IllegalArgumentException("null element found in $this.\")\n }\n }\n
@Suppress("UNCHECKED_CAST")\n return this as List<T>\n}\n\n/**\n * Splits this collection into a list of
lists each not exceeding the given [size].\n *\n * The last list in the resulting list may have fewer elements than the

```

given [size].\n * \n * @param size the number of elements to take in each list, must be positive and can be greater than the number of elements in this collection.\n * \n * @sample

```

samples.collections.Collections.Transformations.chunked\n * \n * @SinceKotlin("1.2")\npublic fun <T>
Iterable<T>.chunked(size: Int): List<List<T>> {\n    return windowed(size, size, partialWindows = true)\n}\n\n/**\n * Splits this collection into several lists each not exceeding the given [size]\n * and applies the given [transform] function to an each.\n * \n * @return list of results of the [transform] applied to an each list.\n * \n * Note that the list passed to the [transform] function is ephemeral and is valid only inside that function.\n * You should not store it or allow it to escape in some way, unless you made a snapshot of it.\n * The last list may have fewer elements than the given [size].\n * \n * @param size the number of elements to take in each list, must be positive and can be greater than the number of elements in this collection.\n * \n * @sample samples.text.Strings.chunkedTransform\n * \n * @SinceKotlin("1.2")\npublic fun <T, R> Iterable<T>.chunked(size: Int, transform: (List<T>) -> R): List<R>
{\n    return windowed(size, size, partialWindows = true, transform = transform)\n}\n\n/**\n * Returns a list containing all elements of the original collection without the first occurrence of the given [element].\n * \n * @public operator fun <T> Iterable<T>.minus(element: T): List<T> {\n    val result =
ArrayList<T>(collectionSizeOrDefault(10))\n    var removed = false\n    return this.filterTo(result) { if (!removed && it == element) { removed = true; false } else true }\n}\n\n/**\n * Returns a list containing all elements of the original collection except the elements contained in the given [elements] array.\n * \n * Before Kotlin 1.6, the [elements] array may have been converted to a [HashSet] to speed up the operation, thus the elements were required to have\n * a correct and stable implementation of `hashCode()` that didn't change between successive invocations.\n * On JVM, you can enable this behavior back with the system property
`kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`.\n * \n * @public operator fun <T>
Iterable<T>.minus(elements: Array<out T>): List<T> {\n    if (elements.isEmpty()) return this.toList()\n    val other = elements.convertToSetForSetOperation()\n    return this.filterNot { it in other }\n}\n\n/**\n * Returns a list containing all elements of the original collection except the elements contained in the given [elements] collection.\n * \n * Before Kotlin 1.6, the [elements] collection may have been converted to a [HashSet] to speed up the operation, thus the elements were required to have\n * a correct and stable implementation of `hashCode()` that didn't change between successive invocations.\n * On JVM, you can enable this behavior back with the system property
`kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`.\n * \n * @public operator fun <T>
Iterable<T>.minus(elements: Iterable<T>): List<T> {\n    val other =
elements.convertToSetForSetOperationWith(this)\n    if (other.isEmpty())\n        return this.toList()\n    return this.filterNot { it in other }\n}\n\n/**\n * Returns a list containing all elements of the original collection except the elements contained in the given [elements] sequence.\n * \n * Before Kotlin 1.6, the [elements] sequence may have been converted to a [HashSet] to speed up the operation, thus the elements were required to have\n * a correct and stable implementation of `hashCode()` that didn't change between successive invocations.\n * On JVM, you can enable this behavior back with the system property
`kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`.\n * \n * @public operator fun <T> Iterable<T>.minus(elements: Sequence<T>): List<T> {\n    val other =
elements.convertToSetForSetOperation()\n    if (other.isEmpty())\n        return this.toList()\n    return this.filterNot { it in other }\n}\n\n/**\n * Returns a list containing all elements of the original collection without the first occurrence of the given [element].\n * \n * @kotl.in.internal.InlineOnly\npublic inline fun <T>
Iterable<T>.minusElement(element: T): List<T> {\n    return minus(element)\n}\n\n/**\n * Splits the original collection into pair of lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n * @sample
samples.collections.Iterables.Operations.partition\n * \n * @public inline fun <T> Iterable<T>.partition(predicate: (T) -> Boolean): Pair<List<T>, List<T>> {\n    val first = ArrayList<T>()\n    val second = ArrayList<T>()\n    for (element in this) {\n        if (predicate(element)) {\n            first.add(element)\n        } else {\n            second.add(element)\n        }\n    }\n    return Pair(first, second)\n}\n\n/**\n * Returns a list containing all elements of the original collection and then the given [element].\n * \n * @public operator fun <T> Iterable<T>.plus(element: T): List<T> {\n    if (this is Collection) return this.plus(element)\n    val result = ArrayList<T>()\n
```

```

result.addAll(this)\n result.add(element)\n return result\n}\n\n/**\n * Returns a list containing all elements of the
original collection and then the given [element].\n *\npublic operator fun <T> Collection<T>.plus(element: T):
List<T> {\n val result = ArrayList<T>(size + 1)\n result.addAll(this)\n result.add(element)\n return
result\n}\n\n/**\n * Returns a list containing all elements of the original collection and then all elements of the given
[elements] array.\n *\npublic operator fun <T> Iterable<T>.plus(elements: Array<out T>): List<T> {\n if (this is
Collection) return this.plus(elements)\n val result = ArrayList<T>()\n result.addAll(this)\n
result.addAll(elements)\n return result\n}\n\n/**\n * Returns a list containing all elements of the original
collection and then all elements of the given [elements] array.\n *\npublic operator fun <T>
Collection<T>.plus(elements: Array<out T>): List<T> {\n val result = ArrayList<T>(this.size + elements.size)\n
result.addAll(this)\n result.addAll(elements)\n return result\n}\n\n/**\n * Returns a list containing all elements
of the original collection and then all elements of the given [elements] collection.\n *\npublic operator fun <T>
Iterable<T>.plus(elements: Iterable<T>): List<T> {\n if (this is Collection) return this.plus(elements)\n val
result = ArrayList<T>()\n result.addAll(this)\n result.addAll(elements)\n return result\n}\n\n/**\n * Returns a
list containing all elements of the original collection and then all elements of the given [elements] collection.\n\n
*\npublic operator fun <T> Collection<T>.plus(elements: Iterable<T>): List<T> {\n if (elements is Collection)
{\n val result = ArrayList<T>(this.size + elements.size)\n result.addAll(this)\n
result.addAll(elements)\n return result\n } else {\n val result = ArrayList<T>(this)\n
result.addAll(elements)\n return result\n }\n}\n\n/**\n * Returns a list containing all elements of the original
collection and then all elements of the given [elements] sequence.\n *\npublic operator fun <T>
Iterable<T>.plus(elements: Sequence<T>): List<T> {\n val result = ArrayList<T>()\n result.addAll(this)\n
result.addAll(elements)\n return result\n}\n\n/**\n * Returns a list containing all elements of the original
collection and then all elements of the given [elements] sequence.\n *\npublic operator fun <T>
Collection<T>.plus(elements: Sequence<T>): List<T> {\n val result = ArrayList<T>(this.size + 10)\n
result.addAll(this)\n result.addAll(elements)\n return result\n}\n\n/**\n * Returns a list containing all elements
of the original collection and then the given [element].\n *\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Iterable<T>.plusElement(element: T): List<T> {\n return plus(element)\n}\n\n/**\n * Returns a list containing all
elements of the original collection and then the given [element].\n *\n@kotlin.internal.InlineOnly\npublic inline fun
<T> Collection<T>.plusElement(element: T): List<T> {\n return plus(element)\n}\n\n/**\n * Returns a list of
snapshots of the window of the given [size]\n * sliding along this collection with the given [step], where each\n *
snapshot is a list.\n * \n * Several last lists may have fewer elements than the given [size].\n * \n * Both [size] and
[step] must be positive and can be greater than the number of elements in this collection.\n * @param size the
number of elements to take in each window\n * @param step the number of elements to move the window forward
by on an each step, by default 1\n * @param partialWindows controls whether or not to keep partial windows in the
end if any,\n * by default `false` which means partial windows won't be preserved\n * \n * @sample
samples.collections.Sequences.Transformations.takeWindows\n *\n@SinceKotlin("1.2")\npublic fun <T>
Iterable<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false): List<List<T>> {\n
checkWindowSizeStep(size, step)\n if (this is RandomAccess && this is List) {\n val thisSize = this.size\n
val resultCapacity = thisSize / step + if (thisSize % step == 0) 0 else 1\n val result =
ArrayList<List<T>>(resultCapacity)\n var index = 0\n while (index in 0 until thisSize) {\n val
windowSize = size.coerceAtMost(thisSize - index)\n if (windowSize < size && !partialWindows) break\n
result.add(List(windowSize) { this[it + index] })\n index += step\n }\n return result\n }\n val
result = ArrayList<List<T>>()\n windowedIterator(iterator(), size, step, partialWindows, reuseBuffer =
false).forEach {\n result.add(it)\n }\n return result\n}\n\n/**\n * Returns a list of results of applying the
given [transform] function to\n * an each list representing a view over the window of the given [size]\n * sliding
along this collection with the given [step].\n * \n * Note that the list passed to the [transform] function is ephemeral
and is valid only inside that function.\n * You should not store it or allow it to escape in some way, unless you made
a snapshot of it.\n * Several last lists may have fewer elements than the given [size].\n * \n * Both [size] and [step]
must be positive and can be greater than the number of elements in this collection.\n * @param size the number of

```

```

elements to take in each window\n * @param step the number of elements to move the window forward by on an
each step, by default 1\n * @param partialWindows controls whether or not to keep partial windows in the end if
any,\n * by default `false` which means partial windows won't be preserved\n * \n * @sample
samples.collections.Sequences.Transformations.averageWindows\n * \n @SinceKotlin("1.2")\n\npublic fun <T, R>
Iterable<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (List<T>) -> R):
List<R> {\n    checkWindowSizeStep(size, step)\n    if (this is RandomAccess && this is List) {\n        val thisSize =
this.size\n        val resultCapacity = thisSize / step + if (thisSize % step == 0) 0 else 1\n        val result =
ArrayList<R>(resultCapacity)\n        val window = MovingSubList(this)\n        var index = 0\n        while (index in 0
until thisSize) {\n            val windowSize = size.coerceAtMost(thisSize - index)\n            if (!partialWindows &&
windowSize < size) break\n            window.move(index, index + windowSize)\n            result.add(transform(window))\n            index += step\n        }\n        return result\n    }\n    val result =
ArrayList<R>()\n    windowedIterator(iterator(), size, step, partialWindows, reuseBuffer = true).forEach {\n
result.add(transform(it))\n    }\n    return result\n}\n\n/**\n * Returns a list of pairs built from the elements of `this`
collection and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n
* @sample samples.collections.Iterables.Operations.zipIterable\n * \n\npublic infix fun <T, R> Iterable<T>.zip(other:
Array<out R>): List<Pair<T, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of values
built from the elements of `this` collection and the [other] array with the same index\n * using the provided
[transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n *
\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n * \n\npublic inline fun <T, R, V>
Iterable<T>.zip(other: Array<out R>, transform: (a: T, b: R) -> V): List<V> {\n    val arraySize = other.size\n    val
list = ArrayList<V>(minOf(collectionSizeOrDefault(10), arraySize))\n    var i = 0\n    for (element in this) {\n        if
(i >= arraySize) break\n        list.add(transform(element, other[i++]))\n    }\n    return list\n}\n\n/**\n * Returns a list
of pairs built from the elements of `this` collection and [other] collection with the same index.\n * The returned list
has length of the shortest collection.\n * \n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*\n\npublic infix fun <T, R> Iterable<T>.zip(other: Iterable<R>): List<Pair<T, R>> {\n    return zip(other) { t1, t2 ->
t1 to t2 }\n}\n\n/**\n * Returns a list of values built from the elements of `this` collection and the [other] collection
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest collection.\n * \n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n * \n\npublic inline fun <T, R, V>
Iterable<T>.zip(other: Iterable<R>, transform: (a: T, b: R) -> V): List<V> {\n    val first = iterator()\n    val second
= other.iterator()\n    val list = ArrayList<V>(minOf(collectionSizeOrDefault(10),
other.collectionSizeOrDefault(10)))\n    while (first.hasNext() && second.hasNext()) {\n
list.add(transform(first.next(), second.next()))\n    }\n    return list\n}\n\n/**\n * Returns a list of pairs of each two
adjacent elements in this collection.\n * \n * \n * The returned list is empty if this collection contains less than two
elements.\n * \n * \n * @sample samples.collections.Collections.Transformations.zipWithNext\n
*\n\n@SinceKotlin("1.2")\n\npublic fun <T> Iterable<T>.zipWithNext(): List<Pair<T, T>> {\n    return zipWithNext
{ a, b -> a to b }\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to
an each pair of two adjacent elements in this collection.\n * \n * \n * The returned list is empty if this collection contains
less than two elements.\n * \n * \n * @sample
samples.collections.Collections.Transformations.zipWithNextToFindDeltas\n * \n\n@SinceKotlin("1.2")\n\npublic
inline fun <T, R> Iterable<T>.zipWithNext(transform: (a: T, b: T) -> R): List<R> {\n    val iterator = iterator()\n    if
(!iterator.hasNext()) return emptyList()\n    val result = mutableListOf<R>()\n    var current = iterator.next()\n
while (iterator.hasNext()) {\n        val next = iterator.next()\n        result.add(transform(current, next))\n        current
= next\n    }\n    return result\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and
using the given [prefix] and [postfix] if supplied.\n * \n * \n * If the collection could be huge, you can specify a non-
negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the
[truncated] string (which defaults to `"...`).\n * \n * \n * @sample
samples.collections.Collections.Transformations.joinTo\n * \n\npublic fun <T, A : Appendable>

```

```

Iterable<T>.joinTo(buffer: A, separator: CharSequence = '\', '\', prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((T) -> CharSequence)? = null): A {
    buffer.append(prefix)
    var count = 0
    for (element in this) {
        if (++count > 1) buffer.append(separator)
        if (limit < 0 || count <= limit) {
            buffer.appendElement(element, transform)
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}

Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to \"...\").

@sample samples.collections.Collections.Transformations.joinToString

public fun <T>
Iterable<T>.joinToString(separator: CharSequence = '\', '\', prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((T) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

Returns this collection as an [Iterable].

@kotlin.internal.InlineOnly
public inline fun <T> Iterable<T>.asIterable():
Iterable<T> {
    return this
}

Creates a [Sequence] instance that wraps the original collection returning its elements when being iterated.

@sample
samples.collections.Sequences.Building.sequenceFromCollection

public fun <T> Iterable<T>.asSequence():
Sequence<T> {
    return Sequence { this.iterator() }
}

Returns an average value of elements in the collection.

@kotlin.jvm.JvmName("averageOfByte")
public fun Iterable<Byte>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

Returns an average value of elements in the collection.

@kotlin.jvm.JvmName("averageOfShort")
public fun
Iterable<Short>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

Returns an average value of elements in the collection.

@kotlin.jvm.JvmName("averageOfInt")
public fun Iterable<Int>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

Returns an average value of elements in the collection.

@kotlin.jvm.JvmName("averageOfLong")
public fun Iterable<Long>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

Returns an average value of elements in the collection.

@kotlin.jvm.JvmName("averageOfFloat")
public fun
Iterable<Float>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

Returns an average value of elements in the collection.

@kotlin.jvm.JvmName("averageOfDouble")
public fun Iterable<Double>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

Returns the sum of all elements in the collection.

@kotlin.jvm.JvmName("sumOfByte")
public fun
Iterable<Byte>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

Returns the sum of all elements in the collection.

@kotlin.jvm.JvmName("sumOfShort")
public fun Iterable<Short>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

Returns the sum of all elements in the collection.

@kotlin.jvm.JvmName("sumOfInt")
public fun Iterable<Int>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

Returns the sum of all elements in the collection.

@kotlin.jvm.JvmName("sumOfLong")
public fun Iterable<Long>.sum(): Long {
    var sum: Long = 0L
    for (element in this) {
        sum += element
    }
    return sum
}

Returns the sum of all elements in the collection.

@kotlin.jvm.JvmName("sumOfFloat")
public fun
Iterable<Float>.sum(): Float {
    var sum: Float = 0.0f
    for (element in this) {
        sum += element
    }
}

```

```

return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n
*\n@kotlin.jvm.JvmName("sumOfDouble")\npublic fun Iterable<Double>.sum(): Double {\n    var sum: Double
= 0.0\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n"/**\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\nimport
kotlin.comparisons.naturalOrder\n\nimport kotlin.random.Random\n\nimport kotlin.js.arrayBufferIsView\n\n/**\n *
Returns the array if it's not `null`, or an empty array otherwise.\n * @sample
samples.collections.Arrays.Usage.arrayOrEmpty\n */\n@kotlin.internal.InlineOnly\n\npublic actual inline fun <T>
Array<out T>?.orEmpty(): Array<out T> = this ?: emptyArray<T>()\n\n/**\n * Returns a *typed* array containing
all of the elements of this collection.\n * \n * Allocates an array of runtime type `T` having its size equal to the size
of this collection\n * and populates the array with the elements of this collection.\n * @sample
samples.collections.Collections.Collections.collectionToTypedArray\n */\n@kotlin.internal.InlineOnly\n\npublic
actual inline fun <T> Collection<T>.toArray(): Array<T> =
toArray(this)\n\n@JsName("copyToArray")\n\n@PublishedApi\n\ninternal fun <T> copyToArray(collection:
Collection<T>): Array<T> {\n    return if (collection.asDynamic().toArray !== undefined)\n        collection.asDynamic().toArray().unsafeCast<Array<T>>()\n    else\n        copyToArrayImpl(collection).unsafeCast<Array<T>>()\n}\n\n@JsName("copyToArrayImpl")\n\ninternal actual fun
copyToArrayImpl(collection: Collection<*>): Array<Any?> {\n    val array = emptyArray<Any?>()\n    val iterator
= collection.iterator()\n    while (iterator.hasNext())\n        array.asDynamic().push(iterator.next())\n    return
array\n}\n\n@JsName("copyToExistingArrayImpl")\n\ninternal actual fun <T> copyToArrayImpl(collection:
Collection<*>, array: Array<T>): Array<T> {\n    if (array.size < collection.size)\n        return
copyToArrayImpl(collection).unsafeCast<Array<T>>()\n    val iterator = collection.iterator()\n    var index = 0\n    while (iterator.hasNext()) {\n        array[index++] = iterator.next().unsafeCast<T>()\n    }\n    if (index < array.size)\n        array[index] = null.unsafeCast<T>()\n    }\n    return array\n}\n\n\n/**\n * Returns an immutable list
containing only the specified object [element].\n */\n\npublic fun <T> listOf(element: T): List<T> =
arrayListOf(element)\n\n@PublishedApi\n\n@SinceKotlin("1.3")\n\n@kotlin.internal.InlineOnly\n\ninternal actual
inline fun <E> buildListInternal(builderAction: MutableList<E>.() -> Unit): List<E> {\n    return
ArrayList<E>().apply(builderAction).build()\n}\n\n@PublishedApi\n\n@SinceKotlin("1.3")\n\n@kotlin.internal.Inlin
eOnly\n\ninternal actual inline fun <E> buildListInternal(capacity: Int, builderAction: MutableList<E>.() -> Unit):
List<E> {\n    checkBuilderCapacity(capacity)\n    return
ArrayList<E>(capacity).apply(builderAction).build()\n}\n\n\n/**\n * Returns an immutable set containing only the
specified object [element].\n */\n\npublic fun <T> setOf(element: T): Set<T> =
hashSetOf(element)\n\n@PublishedApi\n\n@SinceKotlin("1.3")\n\n@kotlin.internal.InlineOnly\n\ninternal actual inline
fun <E> buildSetInternal(builderAction: MutableSet<E>.() -> Unit): Set<E> {\n    return
LinkedHashSet<E>().apply(builderAction).build()\n}\n\n@PublishedApi\n\n@SinceKotlin("1.3")\n\n@kotlin.internal.InlineOnly\n\ninternal actual inline fun <E> buildSetInternal(capacity: Int, builderAction: MutableSet<E>.() -> Unit):
Set<E> {\n    return LinkedHashSet<E>(capacity).apply(builderAction).build()\n}\n\n\n/**\n * Returns an
immutable map, mapping only the specified key to the\n * specified value.\n */\n\npublic fun <K, V> mapOf(pair:
Pair<K, V>): Map<K, V> =
hashMapOf(pair)\n\n@PublishedApi\n\n@SinceKotlin("1.3")\n\n@kotlin.internal.InlineOnly\n\ninternal actual inline
fun <K, V> buildMapInternal(builderAction: MutableMap<K, V>.() -> Unit): Map<K, V> {\n    return
LinkedHashMap<K,
V>().apply(builderAction).build()\n}\n\n@PublishedApi\n\n@SinceKotlin("1.3")\n\n@kotlin.internal.InlineOnly\n\ninte
rnal actual inline fun <K, V> buildMapInternal(capacity: Int, builderAction: MutableMap<K, V>.() -> Unit):
Map<K, V> {\n    return LinkedHashMap<K, V>(capacity).apply(builderAction).build()\n}\n\n\n/**\n * Fills the
list with the provided [value].\n * \n * Each element in the list gets replaced with the [value].\n
*/\n\n@SinceKotlin("1.2")\n\npublic actual fun <T> MutableList<T>.fill(value: T): Unit {\n    for (index in
0..lastIndex) {\n        this[index] = value\n    }\n}\n\n\n/**\n * Randomly shuffles elements in this list.\n * \n * See:

```


https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle#The_modern_algorithm

```
*\n@SinceKotlin("1.2")\npublic actual fun <T> MutableList<T>.shuffle(): Unit = shuffle(Random)\n\n/**\n * Returns a new list with the elements of this list randomly shuffled.\n */\n\n@SinceKotlin("1.2")\npublic actual fun\n<T> Iterable<T>.shuffled(): List<T> = toMutableList().apply { shuffle() }\n\n/**\n * Sorts elements in the list in- place according to their natural sort order.\n * The sort is _stable_. It means that equal elements preserve their\n order relative to each other after sorting.\n */\n\n@sample samples.collections.Collections.Sorting.sortMutableList\n\npublic actual fun <T : Comparable<T>> MutableList<T>.sort(): Unit {\n    collectionsSort(this,\n        naturalOrder())\n}\n\n/**\n * Sorts elements in the list in-place according to the order specified with [comparator].\n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\n\n@sample samples.collections.Collections.Sorting.sortMutableListWith\n\npublic actual fun <T>\nMutableList<T>.sortWith(comparator: Comparator<in T>): Unit {\n    collectionsSort(this,\n        comparator)\n}\n\nprivate fun <T> collectionsSort(list: MutableList<T>, comparator: Comparator<in T>) {\n    if\n        (list.size <= 1) return\n\n    val array = copyToArray(list)\n    sortArrayWith(array, comparator)\n\n    for (i in 0 until\n        array.size) {\n        list[i] = array[i]\n    }\n}\n\ninternal actual fun <T> arrayOfNulls(reference: Array<T>, size: Int):\n    Array<T> {\n    return\n        arrayOfNulls<Any>(size).unsafeCast<Array<T>>()\n}\n\n@SinceKotlin("1.3")\n@PublishedApi\n@JsName("arrayCopy")\ninternal fun <T> arrayCopy(source: Array<out T>, destination: Array<in T>, destinationOffset: Int,\n    startIndex: Int, endIndex: Int) {\n    AbstractList.checkRangeIndexes(startIndex, endIndex, source.size)\n    val\n        rangeSize = endIndex - startIndex\n    AbstractList.checkRangeIndexes(destinationOffset, destinationOffset +\n        rangeSize, destination.size)\n    if (arrayBufferIsView(destination) && arrayBufferIsView(source)) {\n        val\n            subrange = source.asDynamic().subarray(startIndex, endIndex)\n        destination.asDynamic().set(subrange,\n            destinationOffset)\n    } else {\n        if (source !== destination || destinationOffset <= startIndex) {\n            for\n                (index in 0 until rangeSize) {\n                destination[destinationOffset + index] = source[startIndex + index]\n            }\n        } else {\n            for (index in rangeSize - 1 downTo 0) {\n                destination[destinationOffset + index] =\n                    source[startIndex + index]\n            }\n        }\n    }\n}\n\n// no singleton map implementation in js, return map as\nis\n\n@Suppress("NOTHING_TO_INLINE")\ninternal actual inline fun <K, V> Map<K,\n    V>.toSingletonMapOrSelf(): Map<K, V> = this\n\n@Suppress("NOTHING_TO_INLINE")\ninternal actual inline\nfun <K, V> Map<out K, V>.toSingletonMap(): Map<K, V> =\n    this.toMutableMap()\n\n@Suppress("NOTHING_TO_INLINE")\ninternal actual inline fun <T> Array<out\n    T>.copyToArrayOfAny(isVarargs: Boolean): Array<out Any?> =\n    if (isVarargs)\n        // no need to copy vararg\n        array in JS\n        this\n    else\n        this.copyOfOf()\n\n@PublishedApi\ninternal actual fun\ncheckIndexOverflow(index: Int): Int {\n    if (index < 0) {\n        throwIndexOverflow()\n    }\n    return\n        index\n}\n\n@PublishedApi\ninternal actual fun checkCountOverflow(count: Int): Int {\n    if (count < 0) {\n        throwCountOverflow()\n    }\n    return\n        count\n}\n\n/**\n * JS map and set implementations do not make use of\n capacities or load factors.\n */\n\n@PublishedApi\ninternal actual fun mapCapacity(expectedSize: Int) =\n    expectedSize\n\n/**\n * Checks a collection builder function capacity argument.\n * In JS no validation is made in\n Map/Set constructor yet.\n */\n\n@SinceKotlin("1.3")\n@PublishedApi\ninternal fun\ncheckBuilderCapacity(capacity: Int) {\n    require(capacity >= 0) { "capacity must be non-negative." }\n}\n\ninternal actual fun brittleContainsOptimizationEnabled(): Boolean = false\n\n/**\n * Copyright 2010-2018\n JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the\n Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("CollectionsKt")\n\npackage\nkotlin.collections\n\n/**\n * Returns the given iterator itself. This allows to use an instance of iterator in a `for`\n loop.\n */\n\n@sample samples.collections.Iterators.iterator\n\n@kotlin.internal.InlineOnly\npublic inline operator\nfun <T> Iterator<T>.iterator(): Iterator<T> = this\n\n/**\n * Returns an [Iterator] that wraps each element produced\n by the original iterator\n * into an [IndexedValue] containing the index of that element and the element itself.\n */\n\n@sample samples.collections.Iterators.withIndexIterator\n\npublic fun <T> Iterator<T>.withIndex():\n    Iterator<IndexedValue<T>> = IndexingIterator(this)\n\n/**\n * Performs the given [operation] on each element of
```

```

this [Iterator].\n * @sample samples.collections.Iterators.forEachIterator\n *^\npublic inline fun <T>
Iterator<T>.forEach(operation: (T) -> Unit): Unit {\n    for (element in this) operation(element)\n}\n\n/**\n *
Iterator transforming original `iterator` into iterator of [IndexedValue], counting index from zero.\n */\ninternal class
IndexingIterator<out T>(private val iterator: Iterator<T>) : Iterator<IndexedValue<T>> {\n    private var index =
0\n    final override fun hasNext(): Boolean = iterator.hasNext()\n    final override fun next(): IndexedValue<T> =
IndexedValue(checkIndexOverflow(index++), iterator.next())\n}\n\n", "/*\n * Copyright 2010-2022 JetBrains s.r.o.
and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("ComparisonsKt")\n\npackage
kotlin.comparisons\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.*\n\n/**\n * Returns the
greater of two values.\n * \n * If values are equal, returns the first one.\n */\n@SinceKotlin("1.1")\npublic expect
fun <T : Comparable<T>> maxOf(a: T, b: T): T\n\n/**\n * Returns the greater of two values.\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Byte, b: Byte):
Byte\n\n/**\n * Returns the greater of two values.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic
expect inline fun maxOf(a: Short, b: Short): Short\n\n/**\n * Returns the greater of two values.\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Int, b: Int): Int\n\n/**\n
* Returns the greater of two values.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic expect inline
fun maxOf(a: Long, b: Long): Long\n\n/**\n * Returns the greater of two values.\n * \n * If either value is `NaN`,
returns `NaN`.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Float,
b: Float): Float\n\n/**\n * Returns the greater of two values.\n * \n * If either value is `NaN`, returns `NaN`.\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Double, b: Double):
Double\n\n/**\n * Returns the greater of three values.\n * \n * If there are multiple equal maximal values, returns the
first of them.\n */\n@SinceKotlin("1.1")\npublic expect fun <T : Comparable<T>> maxOf(a: T, b: T, c: T):
T\n\n/**\n * Returns the greater of three values.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic
expect inline fun maxOf(a: Byte, b: Byte, c: Byte): Byte\n\n/**\n * Returns the greater of three values.\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Short, b: Short, c:
Short): Short\n\n/**\n * Returns the greater of three values.\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Int, b: Int, c: Int):
Int\n\n/**\n * Returns the greater of three values.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic
expect inline fun maxOf(a: Long, b: Long, c: Long): Long\n\n/**\n * Returns the greater of three values.\n * \n * If
any value is `NaN`, returns `NaN`.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic expect inline
fun maxOf(a: Float, b: Float, c: Float): Float\n\n/**\n * Returns the greater of three values.\n * \n * If any value
is `NaN`, returns `NaN`.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a:
Double, b: Double, c: Double): Double\n\n/**\n * Returns the greater of three values according to the order
specified by the given [comparator].\n * \n * If there are multiple equal maximal values, returns the first of them.\n
*\n@SinceKotlin("1.1")\npublic fun <T> maxOf(a: T, b: T, c: T, comparator: Comparator<in T>): T {\n    return
maxOf(a, maxOf(b, c, comparator), comparator)\n}\n\n/**\n * Returns the greater of two values according to the
order specified by the given [comparator].\n * \n * If values are equal, returns the first one.\n
*\n@SinceKotlin("1.1")\npublic fun <T> maxOf(a: T, b: T, comparator: Comparator<in T>): T {\n    return if
(comparator.compare(a, b) >= 0) a else b\n}\n\n/**\n * Returns the greater of the given values.\n * \n * If there are
multiple equal maximal values, returns the first of them.\n */\n@SinceKotlin("1.4")\npublic expect fun <T :
Comparable<T>> maxOf(a: T, vararg other: T): T\n\n/**\n * Returns the greater of the given values.\n
*\n@SinceKotlin("1.4")\npublic expect fun maxOf(a: Byte, vararg other: Byte): Byte\n\n/**\n * Returns the
greater of the given values.\n */\n@SinceKotlin("1.4")\npublic expect fun maxOf(a: Short, vararg other: Short):
Short\n\n/**\n * Returns the greater of the given values.\n */\n@SinceKotlin("1.4")\npublic expect fun maxOf(a:
Int, vararg other: Int): Int\n\n/**\n * Returns the greater of the given values.\n */\n@SinceKotlin("1.4")\npublic
expect fun maxOf(a: Long, vararg other: Long): Long\n\n/**\n * Returns the greater of the given values.\n * \n * If

```

any value is `NaN`, returns `NaN`.
`@SinceKotlin("1.4")` public expect fun maxOf(a: Float, vararg other: Float): Float
Returns the greater of the given values.
If any value is `NaN`, returns `NaN`.
`@SinceKotlin("1.4")` public expect fun maxOf(a: Double, vararg other: Double): Double
Returns the greater of the given values according to the order specified by the given [comparator].
If there are multiple equal maximal values, returns the first of them.
`@SinceKotlin("1.4")` public fun <T> maxOf(a: T, vararg other: T, comparator: Comparator<in T>): T
{
var max = a
for (e in other) if (comparator.compare(max, e) < 0) max = e
return max
}
Returns the smaller of two values.
If values are equal, returns the first one.
`@SinceKotlin("1.1")` public expect fun <T : Comparable<T>> minOf(a: T, b: T): T
Returns the smaller of two values.
`@SinceKotlin("1.1")` @kotlin.internal.InlineOnly public expect inline fun minOf(a: Byte, b: Byte): Byte
Returns the smaller of two values.
`@SinceKotlin("1.1")` @kotlin.internal.InlineOnly public expect inline fun minOf(a: Short, b: Short): Short
Returns the smaller of two values.
`@SinceKotlin("1.1")` @kotlin.internal.InlineOnly public expect inline fun minOf(a: Int, b: Int): Int
Returns the smaller of two values.
`@SinceKotlin("1.1")` @kotlin.internal.InlineOnly public expect inline fun minOf(a: Long, b: Long): Long
Returns the smaller of two values.
If either value is `NaN`, returns `NaN`.
`@SinceKotlin("1.1")` @kotlin.internal.InlineOnly public expect inline fun minOf(a: Float, b: Float): Float
Returns the smaller of two values.
If either value is `NaN`, returns `NaN`.
`@SinceKotlin("1.1")` @kotlin.internal.InlineOnly public expect inline fun minOf(a: Double, b: Double): Double
Returns the smaller of two values.
If there are multiple equal minimal values, returns the first of them.
`@SinceKotlin("1.1")` public expect fun <T : Comparable<T>> minOf(a: T, b: T, c: T): T
Returns the smaller of three values.
`@SinceKotlin("1.1")` @kotlin.internal.InlineOnly public expect inline fun minOf(a: Byte, b: Byte, c: Byte): Byte
Returns the smaller of three values.
`@SinceKotlin("1.1")` @kotlin.internal.InlineOnly public expect inline fun minOf(a: Short, b: Short, c: Short): Short
Returns the smaller of three values.
`@SinceKotlin("1.1")` @kotlin.internal.InlineOnly public expect inline fun minOf(a: Int, b: Int, c: Int): Int
Returns the smaller of three values.
`@SinceKotlin("1.1")` @kotlin.internal.InlineOnly public expect inline fun minOf(a: Long, b: Long, c: Long): Long
Returns the smaller of three values.
If any value is `NaN`, returns `NaN`.
`@SinceKotlin("1.1")` @kotlin.internal.InlineOnly public expect inline fun minOf(a: Float, b: Float, c: Float): Float
Returns the smaller of three values.
If any value is `NaN`, returns `NaN`.
`@SinceKotlin("1.1")` @kotlin.internal.InlineOnly public expect inline fun minOf(a: Double, b: Double, c: Double): Double
Returns the smaller of three values according to the order specified by the given [comparator].
If there are multiple equal minimal values, returns the first of them.
`@SinceKotlin("1.1")` public fun <T> minOf(a: T, b: T, c: T, comparator: Comparator<in T>): T
{
return minOf(a, minOf(b, c, comparator), comparator)
}
Returns the smaller of two values according to the order specified by the given [comparator].
If values are equal, returns the first one.
`@SinceKotlin("1.1")` public fun <T> minOf(a: T, b: T, comparator: Comparator<in T>): T
{
return if (comparator.compare(a, b) <= 0) a else b
}
Returns the smaller of the given values.
If there are multiple equal minimal values, returns the first of them.
`@SinceKotlin("1.4")` public expect fun <T : Comparable<T>> minOf(a: T, vararg other: T): T
Returns the smaller of the given values.
`@SinceKotlin("1.4")` public expect fun minOf(a: Byte, vararg other: Byte): Byte
Returns the smaller of the given values.
`@SinceKotlin("1.4")` public expect fun minOf(a: Short, vararg other: Short): Short
Returns the smaller of the given values.
`@SinceKotlin("1.4")` public expect fun minOf(a: Int, vararg other: Int): Int
Returns the smaller of the given values.
`@SinceKotlin("1.4")` public expect fun minOf(a: Long, vararg other: Long): Long
Returns the smaller of the given values.
If any value is `NaN`, returns `NaN`.
`@SinceKotlin("1.4")` public expect fun minOf(a: Float, vararg other: Float): Float
Returns the smaller of the given values.
If any value is `NaN`, returns `NaN`.
`@SinceKotlin("1.4")` public expect fun minOf(a: Double, vararg other: Double): Double
Returns

the smaller of the given values according to the order specified by the given [comparator].\n * \n * If there are multiple equal minimal values, returns the first of them.\n * \n * Since Kotlin(1.4)\n public fun <T> minOf(a: T, vararg other: T, comparator: Comparator<in T>): T {\n var min = a\n for (e in other) if (comparator.compare(min, e) > 0) min = e\n return min\n }\n * \n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n

```
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("MapsKt")\n\npackage kotlin.collections\n\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.*\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns the first non-null value produced by [transform] function being applied to entries of this map in iteration order,\n * or throws [NoSuchElementException] if no non-null value was produced.\n * \n * @sample samples.collections.Collections.Transformations.firstNotNullOf\n
```

```
*\n\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Any> Map<out K, V>.firstNotNullOf(transform: (Map.Entry<K, V>) -> R?): R {\n    return firstNotNullOfOrNull(transform) ?: throw NoSuchElementException("No element of the map was transformed to a non-null value.")\n}\n\n/**\n * Returns the first non-null value produced by [transform] function being applied to entries of this map in iteration order,\n * or `null` if no non-null value was produced.\n * \n * @sample samples.collections.Collections.Transformations.firstNotNullOf\n
```

```
*\n\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Any> Map<out K, V>.firstNotNullOfOrNull(transform: (Map.Entry<K, V>) -> R?): R? {\n    for (element in this) {\n        val result = transform(element)\n        if (result != null) {\n            return result\n        }\n    }\n    return null\n}\n\n/**\n * Returns a [List] containing all key-value pairs.\n * \n * public fun <K, V> Map<out K, V>.toList(): List<Pair<K, V>> {\n    if (size == 0)\n        return emptyList()\n    val iterator = entries.iterator()\n    if (!iterator.hasNext())\n        return emptyList()\n    val first = iterator.next()\n    if (!iterator.hasNext())\n        return listOf(first.toPair())\n    val result = ArrayList<Pair<K, V>>(size)\n    result.add(first.toPair())\n    do {\n        result.add(iterator.next().toPair())\n    } while (iterator.hasNext())\n    return result\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being invoked on each entry of original map.\n * \n * \n * @sample samples.collections.Maps.Transformations.flatMap\n
```

```
*\n\npublic inline fun <K, V, R> Map<out K, V>.flatMap(transform: (Map.Entry<K, V>) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being invoked on each entry of original map.\n * \n * \n * @sample samples.collections.Collections.Transformations.flatMap\n
```

```
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapSequence")\npublic inline fun <K, V, R> Map<out K, V>.flatMap(transform: (Map.Entry<K, V>) -> Sequence<R>): List<R> {\n    return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each entry of original map, to the given [destination].\n * \n * public inline fun <K, V, R, C : MutableCollection<in R>> Map<out K, V>.flatMapTo(destination: C, transform: (Map.Entry<K, V>) -> Iterable<R>): C {\n    for (element in this) {\n        val list = transform(element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each entry of original map, to the given [destination].\n
```

```
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapSequenceTo")\npublic inline fun <K, V, R, C : MutableCollection<in R>> Map<out K, V>.flatMapTo(destination: C, transform: (Map.Entry<K, V>) -> Sequence<R>): C {\n    for (element in this) {\n        val list = transform(element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each entry in the original map.\n * \n * \n * @sample samples.collections.Maps.Transformations.mapToList\n
```

```

V>.map(transform: (Map.Entry<K, V> -> R): List<R> {\n  return mapTo(ArrayList<R>(size),
transform)\n}\n\n/**\n * Returns a list containing only the non-null results of applying the given [transform]
function\n * to each entry in the original map.\n * \n * @sample
samples.collections.Maps.Transformations.mapNotNull\n */\npublic inline fun <K, V, R : Any> Map<out K,
V>.mapNotNull(transform: (Map.Entry<K, V> -> R?): List<R> {\n  return mapNotNullTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Applies the given [transform] function to each entry in the original map\n * and appends
only the non-null results to the given [destination].\n */\npublic inline fun <K, V, R : Any, C : MutableCollection<in
R>> Map<out K, V>.mapNotNullTo(destination: C, transform: (Map.Entry<K, V> -> R?): C {\n  forEach {
element -> transform(element)?.let { destination.add(it) } }\n  return destination\n}\n\n/**\n * Applies the given
[transform] function to each entry of the original map\n * and appends the results to the given [destination].\n
*/\npublic inline fun <K, V, R, C : MutableCollection<in R>> Map<out K, V>.mapTo(destination: C, transform:
(Map.Entry<K, V> -> R): C {\n  for (item in this)\n    destination.add(transform(item))\n  return
destination\n}\n\n/**\n * Returns `true` if all entries match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.all\n */\npublic inline fun <K, V> Map<out K, V>.all(predicate:
(Map.Entry<K, V> -> Boolean): Boolean {\n  if (isEmpty()) return true\n  for (element in this) if
(!predicate(element)) return false\n  return true\n}\n\n/**\n * Returns `true` if map has at least one entry.\n * \n *
@sample samples.collections.Collections.Aggregates.any\n */\npublic fun <K, V> Map<out K, V>.any(): Boolean
{\n  return !isEmpty()\n}\n\n/**\n * Returns `true` if at least one entry matches the given [predicate].\n * \n *
@sample samples.collections.Collections.Aggregates.anyWithPredicate\n */\npublic inline fun <K, V> Map<out K,
V>.any(predicate: (Map.Entry<K, V> -> Boolean): Boolean {\n  if (isEmpty()) return false\n  for (element in
this) if (predicate(element)) return true\n  return false\n}\n\n/**\n * Returns the number of entries in this map.\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.count(): Int {\n  return size\n}\n\n/**\n
 * Returns the number of entries matching the given [predicate].\n */\npublic inline fun <K, V> Map<out K,
V>.count(predicate: (Map.Entry<K, V> -> Boolean): Int {\n  if (isEmpty()) return 0\n  var count = 0\n  for
(element in this) if (predicate(element)) ++count\n  return count\n}\n\n/**\n * Performs the given [action] on each
entry.\n */\n@kotlin.internal.HidesMembers\npublic inline fun <K, V> Map<out K, V>.forEach(action:
(Map.Entry<K, V> -> Unit): Unit {\n  for (element in this) action(element)\n}\n\n/**\n * Returns the first entry
yielding the largest value of the given function.\n * \n * @throws NoSuchElementException if the map is empty.\n *
\n * @sample samples.collections.Collections.Aggregates.maxBy\n
*/\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow")\n@kotlin.internal.InlineOnly\n@Suppress
("CONFLICTING_OVERLOADS")\npublic inline fun <K, V, R : Comparable<R>> Map<out K,
V>.maxBy(selector: (Map.Entry<K, V> -> R): Map.Entry<K, V> {\n  return entries.maxBy(selector)\n}\n\n/**\n
 * Returns the first entry yielding the largest value of the given function or `null` if there are no entries.\n * \n *
@sample samples.collections.Collections.Aggregates.maxByOrNull\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out
K, V>.maxByOrNull(selector: (Map.Entry<K, V> -> R): Map.Entry<K, V>? {\n  return
entries.maxByOrNull(selector)\n}\n\n/**\n * Returns the largest value among all values produced by [selector]
function\n * applied to each entry in the map.\n * \n * If any of values produced by [selector] function is `NaN`, the
returned result is `NaN`.\n * \n * @throws NoSuchElementException if the map is empty.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.maxOf(selector:
(Map.Entry<K, V> -> Double): Double {\n  return entries.maxOf(selector)\n}\n\n/**\n * Returns the largest value
among all values produced by [selector] function\n * applied to each entry in the map.\n * \n * If any of values
produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException
if the map is empty.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.maxOf(selector:
(Map.Entry<K, V> -> Float): Float {\n  return entries.maxOf(selector)\n}\n\n/**\n * Returns the largest value

```

among all values produced by [selector] function\n * applied to each entry in the map.\n * \n * @throws NoSuchElementException if the map is empty.\n

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out K,
V>.maxOf(selector: (Map.Entry<K, V>) -> R): R {\n    return entries.maxOf(selector)\n}\n\n/**\n * Returns the
largest value among all values produced by [selector] function\n * applied to each entry in the map or `null` if there
are no entries.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K,
V>.maxOfOrNull(selector: (Map.Entry<K, V>) -> Double): Double? {\n    return
entries.maxOfOrNull(selector)\n}\n\n/**\n * Returns the largest value among all values produced by [selector]
function\n * applied to each entry in the map or `null` if there are no entries.\n * \n * If any of values produced by
[selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K,
V>.maxOfOrNull(selector: (Map.Entry<K, V>) -> Float): Float? {\n    return
entries.maxOfOrNull(selector)\n}\n\n/**\n * Returns the largest value among all values produced by [selector]
function\n * applied to each entry in the map or `null` if there are no entries.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out K,
V>.maxOfOrNull(selector: (Map.Entry<K, V>) -> R): R? {\n    return entries.maxOfOrNull(selector)\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each entry in the map.\n * \n * @throws NoSuchElementException if the map is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R> Map<out K,
V>.maxOfWith(comparator: Comparator<in R>, selector: (Map.Entry<K, V>) -> R): R {\n    return
entries.maxOfWith(comparator, selector)\n}\n\n/**\n * Returns the largest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each entry in the map or `null` if there
are no entries.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R> Map<out K,
V>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Map.Entry<K, V>) -> R): R? {\n    return
entries.maxOfWithOrNull(comparator, selector)\n}\n\n/**\n * Returns the first entry having the largest value
according to the provided [comparator].\n * \n * @throws NoSuchElementException if the map is empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow")\n@kotlin.internal.InlineOnly\n@Suppress(
"CONFLICTING_OVERLOADS")\npublic inline fun <K, V> Map<out K, V>.maxWith(comparator:
Comparator<in Map.Entry<K, V>>): Map.Entry<K, V> {\n    return entries.maxWith(comparator)\n}\n\n/**\n *
Returns the first entry having the largest value according to the provided [comparator] or `null` if there are no
entries.\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K,
V>.maxWithOrNull(comparator: Comparator<in Map.Entry<K, V>>): Map.Entry<K, V>? {\n    return
entries.maxWithOrNull(comparator)\n}\n\n/**\n * Returns the first entry yielding the smallest value of the given
function.\n * \n * @throws NoSuchElementException if the map is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.minBy\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minByOrThrow")\n@kotlin.internal.InlineOnly\n@Suppress(
"CONFLICTING_OVERLOADS")\npublic inline fun <K, V, R : Comparable<R>> Map<out K,
V>.minBy(selector: (Map.Entry<K, V>) -> R): Map.Entry<K, V> {\n    return entries.minBy(selector)\n}\n\n/**\n *
Returns the first entry yielding the smallest value of the given function or `null` if there are no entries.\n * \n *
@sample samples.collections.Collections.Aggregates.minByOrNull\n

```

```

*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out K, V>.minByOrNull(selector: (Map.Entry<K, V>) -> R): Map.Entry<K, V>? {\n return
entries.minByOrNull(selector)\n}\n\n/**\n * Returns the smallest value among all values produced by [selector]
function\n * applied to each entry in the map.\n * \n * If any of values produced by [selector] function is `NaN`, the
returned result is `NaN`.\n * \n * @throws NoSuchElementException if the map is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.minOf(selector:
(Map.Entry<K, V>) -> Double): Double {\n return entries.minOf(selector)\n}\n\n/**\n * Returns the smallest
value among all values produced by [selector] function\n * applied to each entry in the map.\n * \n * If any of values
produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException
if the map is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.minOf(selector:
(Map.Entry<K, V>) -> Float): Float {\n return entries.minOf(selector)\n}\n\n/**\n * Returns the smallest value
among all values produced by [selector] function\n * applied to each entry in the map.\n * \n * @throws
NoSuchElementException if the map is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out K,
V>.minOf(selector: (Map.Entry<K, V>) -> R): R {\n return entries.minOf(selector)\n}\n\n/**\n * Returns the
smallest value among all values produced by [selector] function\n * applied to each entry in the map or `null` if
there are no entries.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K,
V>.minOfOrNull(selector: (Map.Entry<K, V>) -> Double): Double? {\n return
entries.minOfOrNull(selector)\n}\n\n/**\n * Returns the smallest value among all values produced by [selector]
function\n * applied to each entry in the map or `null` if there are no entries.\n * \n * If any of values produced by
[selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K,
V>.minOfOrNull(selector: (Map.Entry<K, V>) -> Float): Float? {\n return
entries.minOfOrNull(selector)\n}\n\n/**\n * Returns the smallest value among all values produced by [selector]
function\n * applied to each entry in the map or `null` if there are no entries.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out K,
V>.minOfOrNull(selector: (Map.Entry<K, V>) -> R): R? {\n return entries.minOfOrNull(selector)\n}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each entry in the map.\n * \n * @throws NoSuchElementException if the map is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R> Map<out K,
V>.minOfWith(comparator: Comparator<in R>, selector: (Map.Entry<K, V>) -> R): R {\n return
entries.minOfWith(comparator, selector)\n}\n\n/**\n * Returns the smallest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each entry in the map or `null` if there
are no entries.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R> Map<out K,
V>.minOfWithOrNull(comparator: Comparator<in R>, selector: (Map.Entry<K, V>) -> R): R? {\n return
entries.minOfWithOrNull(comparator, selector)\n}\n\n/**\n * Returns the first entry having the smallest value
according to the provided [comparator].\n * \n * @throws NoSuchElementException if the map is empty.\n

```

```

*\/n@SinceKotlin("1.7")\/n@kotlin.jvm.JvmName("minWithOrThrow")\/n@kotlin.internal.InlineOnly\/n@Suppress("CONFLICTING_OVERLOADS")\/npublic inline fun <K, V> Map<out K, V>.minWith(comparator: Comparator<in Map.Entry<K, V>>): Map.Entry<K, V> {\/n    return entries.minWith(comparator)\/n}\/n\/n**\/n * Returns the first entry having the smallest value according to the provided [comparator] or `null` if there are no entries.\/n *\/n@SinceKotlin("1.4")\/n@kotlin.internal.InlineOnly\/npublic inline fun <K, V> Map<out K, V>.minWithOrNull(comparator: Comparator<in Map.Entry<K, V>>): Map.Entry<K, V>? {\/n    return entries.minWithOrNull(comparator)\/n}\/n\/n**\/n * Returns `true` if the map has no entries.\/n *\/n * @sample samples.collections.Collections.Aggregates.none\/n *\/npublic fun <K, V> Map<out K, V>.none(): Boolean {\/n    return isEmpty()\/n}\/n\/n**\/n * Returns `true` if no entries match the given [predicate].\/n *\/n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\/n *\/npublic inline fun <K, V> Map<out K, V>.none(predicate: (Map.Entry<K, V>) -> Boolean): Boolean {\/n    if (isEmpty()) return true\/n    for (element in this) if (predicate(element)) return false\/n    return true\/n}\/n\/n**\/n * Performs the given [action] on each entry and returns the map itself afterwards.\/n *\/n@SinceKotlin("1.1")\/npublic inline fun <K, V, M : Map<out K, V>> M.onEach(action: (Map.Entry<K, V>) -> Unit): M {\/n    return apply { for (element in this) action(element) }\/n}\/n\/n**\/n * Performs the given [action] on each entry, providing sequential index with the entry,\/n * and returns the map itself afterwards.\/n * @param [action] function that takes the index of an entry and the entry itself\/n * and performs the action on the entry.\/n *\/n@SinceKotlin("1.4")\/npublic inline fun <K, V, M : Map<out K, V>> M.onEachIndexed(action: (index: Int, Map.Entry<K, V>) -> Unit): M {\/n    return apply { entries.forEachIndexed(action) }\/n}\/n\/n**\/n * Creates an [Iterable] instance that wraps the original map returning its entries when being iterated.\/n *\/n@kotlin.internal.InlineOnly\/npublic inline fun <K, V> Map<out K, V>.asIterable(): Iterable<Map.Entry<K, V>> {\/n    return entries\/n}\/n\/n**\/n * Creates a [Sequence] instance that wraps the original map returning its entries when being iterated.\/n *\/npublic fun <K, V> Map<out K, V>.asSequence(): Sequence<Map.Entry<K, V>> {\/n    return entries.asSequence()\/n}\/n\/n", "/*\/n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\/n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\/n *\/npackage kotlin.text\/n\/n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt\/n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\/n\/n// 10 mappings totally\/ninternal fun Char.titlecaseImpl(): String {\/n    val uppercase = uppercase()\/n    if (uppercase.length > 1) {\/n        return if (this == '\u0149') uppercase else uppercase[0] + uppercase.substring(1).lowercase()\/n    }\/n    return titlecaseChar().toString()\/n}\/n", "/*\/n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\/n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\/n *\/npackage kotlin.text\/n\/n**\/n * Converts this character to lower case using Unicode mapping rules of the invariant locale.\/n *\/n@Deprecated("Use lowercaseChar() instead.")\/nReplaceWith("lowercaseChar()")\/n@DeprecatedSinceKotlin(warningSince = "1.5")\/n@kotlin.internal.InlineOnly\/npublic actual inline fun Char.toLowerCase(): Char = lowercaseChar()\/n\/n**\/n * Converts this character to lower case using Unicode mapping rules of the invariant locale.\/n *\/n * This function performs one-to-one character mapping.\/n * To support one-to-many character mapping use the [lowercase] function.\/n * If this character has no mapping equivalent, the character itself is returned.\/n *\/n * @sample samples.text.Chars.lowercase\/n *\/n@SinceKotlin("1.5")\/n@WasExperimental(ExperimentalStdlibApi::class)\/n@kotlin.internal.InlineOnly\/npublic actual inline fun Char.lowercaseChar(): Char = lowercase()[0]\/n\/n**\/n * Converts this character to lower case using Unicode mapping rules of the invariant locale.\/n *\/n * This function supports one-to-many character mapping, thus the length of the returned string can be greater than one.\/n * For example, ``\u0130'.lowercase()`` returns ``\u0069\u0307``.\/n * where ``\u0130`` is the LATIN CAPITAL LETTER I WITH DOT ABOVE character (`\u0130`).\/n * If this character has no lower case mapping, the result of `toString()` of this char is returned.\/n *\/n * @sample samples.text.Chars.lowercase\/n *\/n@SinceKotlin("1.5")\/n@WasExperimental(ExperimentalStdlibApi::class)\/n@kotlin.internal.InlineOnly\/npublic actual inline fun Char.lowercase(): String = toString().asDynamic().toLowerCase().unsafeCast<String>()\/n\/n**\/n

```



```

Unicode Standard.\n *\n * @sample samples.text.Chars.isUpperCase\n *\n @SinceKotlin("1.5")\npublic actual fun
Char.isUpperCase(): Boolean {\n if (this in 'A'..'Z') {\n return true\n }\n if (this < "\u0080") {\n return
false\n }\n return isUpperCaseImpl()\n}\n\n/**\n * Returns `true` if this character is lower case.\n *\n * A
character is considered to be a lower case character if its [category] is [CharCategory.LOWERCASE_LETTER],\n *\n
or it has contributory property `Other_Lowercase` as defined by the Unicode Standard.\n *\n * @sample
samples.text.Chars.isLowerCase\n *\n @SinceKotlin("1.5")\npublic actual fun Char.isLowerCase(): Boolean {\n
if (this in 'a'..'z') {\n return true\n }\n if (this < "\u0080") {\n return false\n }\n return
isLowerCaseImpl()\n}\n\n/**\n * Returns `true` if this character is a title case letter.\n *\n * A character is
considered to be a title case letter if its [category] is [CharCategory.TITLECASE_LETTER].\n *\n * @sample
samples.text.Chars.isTitleCase\n *\n @SinceKotlin("1.5")\npublic actual fun Char.isTitleCase(): Boolean {\n if
(this < "\u0080") {\n return false\n }\n return getCategoryValue() ==
CharCategory.TITLECASE_LETTER.value\n}\n\n/**\n * Returns `true` if this character is an ISO control
character.\n *\n * A character is considered to be an ISO control character if its [category] is
[CharCategory.CONTROL],\n *\n meaning the Char is in the range "\u0000'..\u001F" or in the range
"\u007F'..\u009F".\n *\n * @sample samples.text.Chars.isISOControl\n *\n @SinceKotlin("1.5")\npublic actual
fun Char.isISOControl(): Boolean {\n return this <= "\u001F" || this in "\u007F'..\u009F"\n}\n\n/**\n *
Determines whether a character is whitespace according to the Unicode standard.\n *\n * Returns `true` if the character
is whitespace.\n *\n * @sample samples.text.Chars.isWhitespace\n *\n @SinceKotlin("1.5")\npublic actual fun
Char.isWhitespace(): Boolean = isWhitespaceImpl()\n}\n\n/* Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n *\n @package kotlin.text\n @import kotlin.js.RegExp\n\n/**\n * Converts the characters
in the specified array to a string.\n *\n @SinceKotlin("1.2")\n @Deprecated("Use CharArray.concatToString()
instead", ReplaceWith("chars.concatToString()"))\n @DeprecatedSinceKotlin(warningSince = "1.4", errorSince
= "1.5")\npublic actual fun String(chars: CharArray): String {\n var result = ""\n for (char in chars) {\n
result += char\n }\n return result\n}\n\n/**\n * Converts the characters from a portion of the specified array to a
string.\n *\n * @throws IndexOutOfBoundsException if either [offset] or [length] are less than zero\n *\n or `offset +
length` is out of [chars] array bounds.\n *\n @SinceKotlin("1.2")\n @Deprecated("Use
CharArray.concatToString(startIndex, endIndex) instead", ReplaceWith("chars.concatToString(offset, offset +
length)"))\n @DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5")\npublic actual fun String(chars:
CharArray, offset: Int, length: Int): String {\n if (offset < 0 || length < 0 || chars.size - offset < length)\n
throw IndexOutOfBoundsException("size: ${chars.size}; offset: $offset; length: $length")\n var result = ""\n
for (index in offset until offset + length) {\n result += chars[index]\n }\n return result\n}\n\n/**\n *
Concatenates characters in this [CharArray] into a String.\n *\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
CharArray.concatToString(): String {\n var result = ""\n for (char in this) {\n result += char\n }\n
return result\n}\n\n/**\n * Concatenates characters in this [CharArray] or its subrange into a String.\n *\n * @param
startIndex the beginning (inclusive) of the subrange of characters, 0 by default.\n *\n * @param endIndex the end
(exclusive) of the subrange of characters, size of this array by default.\n *\n * @throws
IndexOutOfBoundsException if [startIndex] is less than zero or [endIndex] is greater than the size of this array.\n *\n
* @throws IllegalArgumentException if [startIndex] is greater than [endIndex].\n *\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n @Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun CharArray.concatToString(startIndex: Int = 0,
endIndex: Int = this.size): String {\n AbstractList.checkBoundsIndexes(startIndex, endIndex, this.size)\n var
result = ""\n for (index in startIndex until endIndex) {\n result += this[index]\n }\n return
result\n}\n\n/**\n * Returns a [CharArray] containing characters of this string.\n *\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
String.toCharArray(): CharArray {\n return CharArray(length) { get(it) }\n}\n\n/**\n * Returns a [CharArray]
containing characters of this string or its substring.\n *\n * @param startIndex the beginning (inclusive) of the

```

```

substring, 0 by default.\n * @param endIndex the end (exclusive) of the substring, length of this string by default.\n
*\n * @throws IndexOutOfBoundsException if [startIndex] is less than zero or [endIndex] is greater than the length
of this string.\n * @throws IllegalArgumentException if [startIndex] is greater than [endIndex].\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun String.toCharArray(startIndex: Int = 0, endIndex: Int
= this.length): CharArray {\n    AbstractList.checkBoundsIndexes(startIndex, endIndex, length)\n    return
CharArray(endIndex - startIndex) { get(startIndex + it) }\n}\n\n/**\n * Decodes a string from the bytes in UTF-8
encoding in this array.\n * Malformed byte sequences are replaced by the replacement char `\\uFFFD`.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
ByteArray.decodeToString(): String {\n    return decodeUtf8(this, 0, size, false)\n}\n\n/**\n * Decodes a string from
the bytes in UTF-8 encoding in this array or its subrange.\n * @param startIndex the beginning (inclusive) of the
subrange to decode, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to decode, size of this
array by default.\n * @param throwOnInvalidSequence specifies whether to throw an exception on malformed byte
sequence or replace it by the replacement char `\\uFFFD`.\n * @throws IndexOutOfBoundsException if
[startIndex] is less than zero or [endIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [startIndex] is greater than [endIndex].\n * @throws CharacterCodingException if the
byte array contains malformed UTF-8 byte sequence and [throwOnInvalidSequence] is true.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun ByteArray.decodeToString(\n    startIndex: Int = 0,\n    endIndex: Int = this.size,\n    throwOnInvalidSequence: Boolean = false\n): String {\n
    AbstractList.checkBoundsIndexes(startIndex, endIndex, this.size)\n    return decodeUtf8(this, startIndex, endIndex,
throwOnInvalidSequence)\n}\n\n/**\n * Encodes this string to an array of bytes in UTF-8 encoding.\n * Any
malformed char sequence is replaced by the replacement byte sequence.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
String.encodeToByteArray(): ByteArray {\n    return encodeUtf8(this, 0, length, false)\n}\n\n/**\n * Encodes this
string or its substring to an array of bytes in UTF-8 encoding.\n * @param startIndex the beginning (inclusive)
of the substring to encode, 0 by default.\n * @param endIndex the end (exclusive) of the substring to encode, length
of this string by default.\n * @param throwOnInvalidSequence specifies whether to throw an exception on
malformed char sequence or replace.\n * @throws IndexOutOfBoundsException if [startIndex] is less than zero
or [endIndex] is greater than the length of this string.\n * @throws IllegalArgumentException if [startIndex] is
greater than [endIndex].\n * @throws CharacterCodingException if this string contains malformed char sequence
and [throwOnInvalidSequence] is true.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun String.encodeToByteArray(\n    startIndex: Int = 0,\n    endIndex: Int = this.length,\n    throwOnInvalidSequence: Boolean = false\n): ByteArray {\n
    AbstractList.checkBoundsIndexes(startIndex, endIndex, length)\n    return encodeUtf8(this, startIndex, endIndex,
throwOnInvalidSequence)\n}\n\n/**\n * Returns a copy of this string converted to upper case using the rules of the
default locale.\n * @n@Deprecated("Use uppercase() instead.")\n    ReplaceWith("uppercase()")\n@DeprecatedSinceKotlin(warningSince =
"1.5")\n@kotlin.internal.InlineOnly\npublic actual inline fun String.toUpperCase(): String =
asDynamic().toUpperCase()\n\n/**\n * Returns a copy of this string converted to upper case using Unicode mapping
rules of the invariant locale.\n * This function supports one-to-many and many-to-one character mapping,\n *
thus the length of the returned string can be different from the length of the original string.\n * @sample
samples.text.Strings.uppercase\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
c actual inline fun String.toUpperCase(): String = asDynamic().toUpperCase()\n\n/**\n * Returns a copy of this string
converted to lower case using the rules of the default locale.\n * @n@Deprecated("Use lowercase() instead.")\n    ReplaceWith("lowercase()")\n@DeprecatedSinceKotlin(warningSince =

```



```

the specified [other], optionally ignoring case difference.\n * \n * @param ignoreCase `true` to ignore character case
when comparing contents.\n * \n * @sample samples.text.Strings.contentEquals\n * \n * @SinceKotlin("1.5")\n\npublic
actual fun CharSequence?.contentEquals(other: CharSequence?, ignoreCase: Boolean): Boolean {\n    return if
(ignoreCase)\n        this.contentEqualsIgnoreCaseImpl(other)\n    else\n
this.contentEqualsImpl(other)\n}\n\nprivate val STRING_CASE_INSENSITIVE_ORDER = Comparator<String>
{ a, b -> a.compareTo(b, ignoreCase = true) }\n\n@SinceKotlin("1.2")\n\npublic actual val
String.Companion.CASE_INSENSITIVE_ORDER: Comparator<String>\n    get() =
STRING_CASE_INSENSITIVE_ORDER\n", "/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("CharsKt")\n\npackage kotlin.text\n\n/**\n
* Returns the numeric value of the decimal digit that this Char represents.\n * Throws an exception if this Char is
not a valid decimal digit.\n * \n * A Char is considered to represent a decimal digit if [isDigit] is true for the Char.\n
* In this case, the Unicode decimal digit value of the character is returned.\n * \n * @sample
samples.text.Chars.digitToInt\n
*\n\n@SinceKotlin("1.5")\n\n@WasExperimental(ExperimentalStdlibApi::class)\n\npublic fun Char.digitToInt(): Int
{\n    return digitOf(this, 10).also {\n        if (it < 0) throw IllegalArgumentException("Char $this is not a decimal
digit")\n    }\n}\n\n/**\n
* Returns the numeric value of the digit that this Char represents in the specified [radix].\n * Throws an exception if the [radix] is not in the range `2..36` or if this Char is not a valid digit in the specified
[radix].\n * \n * A Char is considered to represent a digit in the specified [radix] if at least one of the following is
true:\n * - [isDigit] is `true` for the Char and the Unicode decimal digit value of the character is less than the
specified [radix]. In this case the decimal digit value is returned.\n * - The Char is one of the uppercase Latin letters
'A' through 'Z' and its [code] is less than `radix + 'A'.code - 10`. In this case, `this.code - 'A'.code + 10` is returned.\n
* - The Char is one of the lowercase Latin letters 'a' through 'z' and its [code] is less than `radix + 'a'.code - 10`. In
this case, `this.code - 'a'.code + 10` is returned.\n * - The Char is one of the fullwidth Latin capital letters '\uFF21'
through '\uFF3A' and its [code] is less than `radix + 0xFF21 - 10`. In this case, `this.code - 0xFF21 + 10` is
returned.\n * - The Char is one of the fullwidth Latin small letters '\uFF41' through '\uFF5A' and its [code] is less
than `radix + 0xFF41 - 10`. In this case, `this.code - 0xFF41 + 10` is returned.\n * \n * @sample
samples.text.Chars.digitToInt\n
*\n\n@SinceKotlin("1.5")\n\n@WasExperimental(ExperimentalStdlibApi::class)\n\npublic fun Char.digitToInt(radix:
Int): Int {\n    return digitToIntOrNull(radix) ?: throw IllegalArgumentException("Char $this is not a digit in the
given radix=$radix")\n}\n\n/**\n
* Returns the numeric value of the decimal digit that this Char represents, or
`null` if this Char is not a valid decimal digit.\n * \n * A Char is considered to represent a decimal digit if [isDigit] is
true for the Char.\n * In this case, the Unicode decimal digit value of the character is returned.\n * \n * @sample
samples.text.Chars.digitToIntOrNull\n
*\n\n@SinceKotlin("1.5")\n\n@WasExperimental(ExperimentalStdlibApi::class)\n\npublic fun
Char.digitToIntOrNull(): Int? {\n    return digitOf(this, 10).takeIf { it >= 0 }\n}\n\n/**\n
* Returns the numeric
value of the digit that this Char represents in the specified [radix], or `null` if this Char is not a valid digit in the
specified [radix].\n * Throws an exception if the [radix] is not in the range `2..36`.\n * \n * A Char is considered to
represent a digit in the specified [radix] if at least one of the following is true:\n * - [isDigit] is `true` for the Char
and the Unicode decimal digit value of the character is less than the specified [radix]. In this case the decimal digit
value is returned.\n * - The Char is one of the uppercase Latin letters 'A' through 'Z' and its [code] is less than `radix
+ 'A'.code - 10`. In this case, `this.code - 'A'.code + 10` is returned.\n * - The Char is one of the lowercase Latin
letters 'a' through 'z' and its [code] is less than `radix + 'a'.code - 10`. In this case, `this.code - 'a'.code + 10` is
returned.\n * - The Char is one of the fullwidth Latin capital letters '\uFF21' through '\uFF3A' and its [code] is less
than `radix + 0xFF21 - 10`. In this case, `this.code - 0xFF21 + 10` is returned.\n * - The Char is one of the fullwidth
Latin small letters '\uFF41' through '\uFF5A' and its [code] is less than `radix + 0xFF41 - 10`. In this case,
`this.code - 0xFF41 + 10` is returned.\n * \n * @sample samples.text.Chars.digitToIntOrNull\n

```

```

*^@SinceKotlin("1.5")^@WasExperimental(ExperimentalStdlibApi::class)^npublic fun
Char.digitToIntOrNull(radix: Int): Int? {^n checkRadix(radix)^n return digitOf(this, radix).takeIf { it >= 0
}^n}^n/^n/^n * Returns the Char that represents this decimal digit.^n * Throws an exception if this value is not in the
range `0..9`.^n * If this value is in `0..9`, the decimal digit Char with code `0.code + this` is returned.^n *^n *
@sample samples.text.Chars.digitToChar^n
*^@SinceKotlin("1.5")^@WasExperimental(ExperimentalStdlibApi::class)^npublic fun Int.digitToChar(): Char
{^n if (this in 0..9) {^n return '0' + this^n }^n throw IllegalArgumentException("Int $this is not a decimal
digit")^n}^n/^n/^n * Returns the Char that represents this numeric digit value in the specified [radix].^n * Throws
an exception if the [radix] is not in the range `2..36` or if this value is not in the range `0 until radix`.^n * If this
value is less than `10`, the decimal digit Char with code `0.code + this` is returned.^n * Otherwise, the uppercase
Latin letter with code `A.code + this - 10` is returned.^n *^n * @sample samples.text.Chars.digitToChar^n
*^@SinceKotlin("1.5")^@WasExperimental(ExperimentalStdlibApi::class)^npublic fun Int.digitToChar(radix:
Int): Char {^n if (radix !in 2..36) {^n throw IllegalArgumentException("Invalid radix: $radix. Valid radix
values are in range 2..36")^n }^n if (this < 0 || this >= radix) {^n throw IllegalArgumentException("Digit
$this does not represent a valid digit in radix $radix")^n }^n return if (this < 10) {^n '0' + this^n } else {^n
'A' + this - 10^n }^n}^n/^n/^n * Converts this character to lower case using Unicode mapping rules of the
invariant locale.^n *^n * @Deprecated("Use lowercaseChar() instead."),
ReplaceWith("lowercaseChar()")^n@DeprecatedSinceKotlin(warningSince = "1.5")^npublic expect fun
Char.toLowerCase(): Char^n/^n/^n * Converts this character to lower case using Unicode mapping rules of the
invariant locale.^n *^n * This function performs one-to-one character mapping.^n * To support one-to-many
character mapping use the [lowercase] function.^n * If this character has no mapping equivalent, the character itself
is returned.^n *^n * @sample samples.text.Chars.toLowerCase^n
*^@SinceKotlin("1.5")^@WasExperimental(ExperimentalStdlibApi::class)^npublic expect fun
Char.lowercaseChar(): Char^n/^n/^n * Converts this character to lower case using Unicode mapping rules of the
invariant locale.^n *^n * This function supports one-to-many character mapping, thus the length of the returned
string can be greater than one.^n * For example, ``\u0130'.lowercase()`` returns ``\u0069\u0307``,^n * where
``\u0130`` is the LATIN CAPITAL LETTER I WITH DOT ABOVE character (``\ufffd\ufffd``).^n * If this character
has no lower case mapping, the result of `toString()` of this char is returned.^n *^n * @sample
samples.text.Chars.lowercase^n
*^@SinceKotlin("1.5")^@WasExperimental(ExperimentalStdlibApi::class)^npublic expect fun
Char.lowercase(): String^n/^n/^n * Converts this character to upper case using Unicode mapping rules of the
invariant locale.^n *^n * @Deprecated("Use uppercaseChar() instead."),
ReplaceWith("uppercaseChar()")^n@DeprecatedSinceKotlin(warningSince = "1.5")^npublic expect fun
Char.toUpperCase(): Char^n/^n/^n * Converts this character to upper case using Unicode mapping rules of the
invariant locale.^n *^n * This function performs one-to-one character mapping.^n * To support one-to-many
character mapping use the [uppercase] function.^n * If this character has no mapping equivalent, the character itself
is returned.^n *^n * @sample samples.text.Chars.uppercase^n
*^@SinceKotlin("1.5")^@WasExperimental(ExperimentalStdlibApi::class)^npublic expect fun
Char.uppercaseChar(): Char^n/^n/^n * Converts this character to upper case using Unicode mapping rules of the
invariant locale.^n *^n * This function supports one-to-many character mapping, thus the length of the returned
string can be greater than one.^n * For example, ``\uFB00'.uppercase()`` returns ``\u0046\u0046``,^n * where
``\uFB00`` is the LATIN SMALL LIGATURE FF character (``\ufffd\ufffd\ufffd``).^n * If this character has no upper
case mapping, the result of `toString()` of this char is returned.^n *^n * @sample samples.text.Chars.uppercase^n
*^@SinceKotlin("1.5")^@WasExperimental(ExperimentalStdlibApi::class)^npublic expect fun
Char.uppercase(): String^n/^n/^n * Converts this character to title case using Unicode mapping rules of the
invariant locale.^n *^n * This function performs one-to-one character mapping.^n * To support one-to-many
character mapping use the [titlecase] function.^n * If this character has no mapping equivalent, the result of calling
[uppercaseChar] is returned.^n *^n * @sample samples.text.Chars.titlecase^n *^@SinceKotlin("1.5")^npublic

```

```

expect fun Char.titlecaseChar(): Char\n\n/**\n * Converts this character to title case using Unicode mapping rules of
the invariant locale.\n * This function supports one-to-many character mapping, thus the length of the returned
string can be greater than one.\n * For example, `'\uFB00'.titlecase()` returns `'\u0046\u0066'\n` where
`\uFB00` is the LATIN SMALL LIGATURE FF character (`\ufffd\uuffd\uuffd`).\n * If this character has no title
case mapping, the result of [uppercase] is returned instead.\n *\n * @sample samples.text.Chars.titlecase\n
*/\n\n@SinceKotlin("1.5")\npublic fun Char.titlecase(): String = titlecaseImpl()\n\n/**\n * Concatenates this Char
and a String.\n *\n * @sample samples.text.Chars.plus\n */\n\n@kotlin.internal.InlineOnly\npublic inline operator fun
Char.plus(other: String): String = this.toString() + other\n\n/**\n * Returns `true` if this character is equal to the
[other] character, optionally ignoring character case.\n *\n * Two characters are considered equal ignoring case if
`Char.uppercaseChar().lowercaseChar()` on each character produces the same result.\n *\n * @param ignoreCase
`true` to ignore character case when comparing characters. By default `false`.\n *\n * @sample
samples.text.Chars.equals\n */\n\npublic fun Char.equals(other: Char, ignoreCase: Boolean = false): Boolean {\n    if
(this == other) return true\n    if (!ignoreCase) return false\n    val thisUpper = this.uppercaseChar()\n    val
otherUpper = other.uppercaseChar()\n    return thisUpper == otherUpper || thisUpper.lowercaseChar() ==
otherUpper.lowercaseChar()\n}\n\n/**\n * Returns `true` if this character is a Unicode surrogate code unit.\n
*/\n\npublic fun Char.isSurrogate(): Boolean = this in Char.MIN_SURROGATE..Char.MAX_SURROGATE\n\n/**\n
* Returns the Unicode general category of this character.\n *\n */\n\n@SinceKotlin("1.5")\npublic expect val
Char.category: CharCategory\n\n/**\n * Returns `true` if this character (Unicode code point) is defined in
Unicode.\n *\n * A character is considered to be defined in Unicode if its [category] is not
[CharCategory.UNASSIGNED].\n *\n */\n\n@SinceKotlin("1.5")\npublic expect fun Char.isDefined():
Boolean\n\n/**\n * Returns `true` if this character is a letter.\n *\n * A character is considered to be a letter if its
[category] is [CharCategory.UPPERCASE_LETTER],\n * [CharCategory.LOWERCASE_LETTER],
[CharCategory.TITLECASE_LETTER], [CharCategory.MODIFIER_LETTER], or
[CharCategory.OTHER_LETTER].\n *\n * @sample samples.text.Chars.isLetter\n
*/\n\n@SinceKotlin("1.5")\npublic expect fun Char.isLetter(): Boolean\n\n/**\n * Returns `true` if this character is a
letter or digit.\n *\n * @see isLetter\n * @see isDigit\n *\n * @sample samples.text.Chars.isLetterOrDigit\n
*/\n\n@SinceKotlin("1.5")\npublic expect fun Char.isLetterOrDigit(): Boolean\n\n/**\n * Returns `true` if this
character is a digit.\n *\n * A character is considered to be a digit if its [category] is
[CharCategory.DECIMAL_DIGIT_NUMBER].\n *\n * @sample samples.text.Chars.isDigit\n
*/\n\n@SinceKotlin("1.5")\npublic expect fun Char.isDigit(): Boolean\n\n/**\n * Returns `true` if this character is
upper case.\n *\n * A character is considered to be an upper case character if its [category] is
[CharCategory.UPPERCASE_LETTER],\n * or it has contributory property `Other_Uppercase` as defined by the
Unicode Standard.\n *\n * @sample samples.text.Chars.isUpperCase\n */\n\n@SinceKotlin("1.5")\npublic expect
fun Char.isUpperCase(): Boolean\n\n/**\n * Returns `true` if this character is lower case.\n *\n * A character is
considered to be a lower case character if its [category] is [CharCategory.LOWERCASE_LETTER],\n * or it has
contributory property `Other_Lowercase` as defined by the Unicode Standard.\n *\n * @sample
samples.text.Chars.isLowerCase\n */\n\n@SinceKotlin("1.5")\npublic expect fun Char.isLowerCase():
Boolean\n\n/**\n * Returns `true` if this character is a title case letter.\n *\n * A character is considered to be a title
case letter if its [category] is [CharCategory.TITLECASE_LETTER].\n *\n * @sample
samples.text.Chars.isTitleCase\n */\n\n@SinceKotlin("1.5")\npublic expect fun Char.isTitleCase(): Boolean\n\n/**\n
* Returns `true` if this character is an ISO control character.\n *\n * A character is considered to be an ISO control
character if its [category] is [CharCategory.CONTROL],\n * meaning the Char is in the range `'\u0000'..' \u001F`
or in the range `'\u007F'..' \u009F`.\n *\n * @sample samples.text.Chars.isISOControl\n
*/\n\n@SinceKotlin("1.5")\npublic expect fun Char.isISOControl(): Boolean\n\n/**\n * Determines whether a
character is whitespace according to the Unicode standard.\n * Returns `true` if the character is whitespace.\n *\n
*/\n\n@sample samples.text.Chars.isWhitespace\n */\n\npublic expect fun Char.isWhitespace(): Boolean\n\n"/**\n
* Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage

```

```

kotlin\n\n\n/**\n * Creates a Char with the specified [code], or throws an exception if the [code] is out of
`Char.MIN_VALUE.code..Char.MAX_VALUE.code`.\n *\n * If the program that calls this function is written in a
way that only valid [code] is passed as the argument,\n * using the overload that takes a [UShort] argument is
preferable (`Char(intValue.toUShort())`).\n * That overload doesn't check validity of the argument, and may
improve program performance when the function is called routinely inside a loop.\n *\n * @sample
samples.text.Chars.charFromCode\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
c inline fun Char(code: Int): Char {\n    if (code < Char.MIN_VALUE.code || code > Char.MAX_VALUE.code) {\n
        throw IllegalArgumentException("Invalid Char code: $code")\n    }\n    return code.toChar()\n}\n\n/**\n *
Creates a Char with the specified [code].\n *\n * @sample samples.text.Chars.charFromCode\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("NO_ACTUAL_FOR
_EXPECT")\npublic expect fun Char(code: UShort): Char\n\n/**\n * Returns the code of this Char.\n *\n * Code of
a Char is the value it was constructed with, and the UTF-16 code unit corresponding to this Char.\n *\n * @sample
samples.text.Chars.code\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\n@Su
ppress("DEPRECATION")\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline val Char.code: Int get() =
this.toInt()\n",/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use
of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("SequencesKt")\n\npackage
kotlin.sequences\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.*\n\n/**\n * Returns
`true` if [element] is found in the sequence.\n *\n * The operation is _terminal_.\n *\n@kotlin.internal.OnlyInputTypes T> Sequence<T>.contains(element: T): Boolean {\n    return indexOf(element)
>= 0\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the
[index] is out of bounds of this sequence.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Elements.elementAt\n *\n@kotlin.internal.OnlyInputTypes T> Sequence<T>.elementAt(index: Int): T
{\n    return elementAtOrElse(index) { throw IndexOutOfBoundsException("Sequence doesn't contain element at
index $index.") }\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue]
function if the [index] is out of bounds of this sequence.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n *\n@kotlin.internal.OnlyInputTypes T> Sequence<T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {\n    if (index < 0)\n        return
defaultValue(index)\n    val iterator = iterator()\n    var count = 0\n    while (iterator.hasNext()) {\n        val element
= iterator.next()\n        if (index == count++)\n            return element\n    }\n    return
defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of
this sequence.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n *\n@kotlin.internal.OnlyInputTypes T> Sequence<T>.elementAtOrNull(index: Int): T? {\n    if (index < 0)\n        return null\n    val iterator = iterator()\n
    var count = 0\n    while (iterator.hasNext()) {\n        val element = iterator.next()\n        if (index == count++)\n            return element\n    }\n    return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null`
if no such element was found.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Elements.find\n *\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Sequence<T>.find(predicate: (T) -> Boolean): T? {\n    return firstOrNull(predicate)\n}\n\n/**\n * Returns the last
element matching the given [predicate], or `null` if no such element was found.\n *\n * The operation is
_terminal_.\n *\n * @sample samples.collections.Collections.Elements.find\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.findLast(predicate: (T) -> Boolean): T? {\n
    return lastOrNull(predicate)\n}\n\n/**\n * Returns the first element.\n *\n * The operation is _terminal_.\n *\n *
@throws NoSuchElementException if the sequence is empty.\n *\n@kotlin.internal.OnlyInputTypes T> Sequence<T>.first(): T {\n    val
iterator = iterator()\n    if (!iterator.hasNext())\n        throw NoSuchElementException("Sequence is empty.")\n}

```



```

return iterator.next()\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws
[NoSuchElementException] if no such element is found.\n *\n * The operation is _terminal_.\n */\npublic inline fun
<T> Sequence<T>.first(predicate: (T) -> Boolean): T {\n    for (element in this) if (predicate(element)) return
element\n    throw NoSuchElementException("Sequence contains no element matching the predicate.")\n}\n\n/**\n * Returns the first non-null value produced by [transform] function being applied to elements of this sequence in
iteration order,\n * or throws [NoSuchElementException] if no non-null value was produced.\n *\n * The operation
is _terminal_.\n *\n * @sample samples.collections.Collections.Transformations.firstNotNullOf\n
*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any>
Sequence<T>.firstNotNullOf(transform: (T) -> R?): R {\n    return firstNotNullOfOrNull(transform) ?: throw
NoSuchElementException("No element of the sequence was transformed to a non-null value.")\n}\n\n/**\n * Returns the first non-null value produced by [transform] function being applied to elements of this sequence in
iteration order,\n * or `null` if no non-null value was produced.\n *\n * The operation is _terminal_.\n *\n *
@sample samples.collections.Collections.Transformations.firstNotNullOf\n
*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any>
Sequence<T>.firstNotNullOfOrNull(transform: (T) -> R?): R? {\n    for (element in this) {\n        val result =
transform(element)\n        if (result != null) {\n            return result\n        }\n    }\n    return null\n}\n\n/**\n * Returns the first element, or `null` if the sequence is empty.\n *\n * The operation is _terminal_.\n */\npublic fun
<T> Sequence<T>.firstOrNull(): T? {\n    val iterator = iterator()\n    if (!iterator.hasNext())\n        return null\n    return iterator.next()\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was
not found.\n *\n * The operation is _terminal_.\n */\npublic inline fun <T> Sequence<T>.firstOrNull(predicate: (T)
-> Boolean): T? {\n    for (element in this) if (predicate(element)) return element\n    return null\n}\n\n/**\n * Returns first index of [element], or -1 if the sequence does not contain element.\n *\n * The operation is
_terminal_.\n */\npublic fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.indexOf(element: T): Int {\n    var
index = 0\n    for (item in this) {\n        checkIndexOverflow(index)\n        if (element == item)\n            return
index\n        index++\n    }\n    return -1\n}\n\n/**\n * Returns index of the first element matching the given
[predicate], or -1 if the sequence does not contain such element.\n *\n * The operation is _terminal_.\n */\npublic
inline fun <T> Sequence<T>.indexOfFirst(predicate: (T) -> Boolean): Int {\n    var index = 0\n    for (item in this)
{\n        checkIndexOverflow(index)\n        if (predicate(item))\n            return index\n        index++\n    }\n    return
-1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the sequence does not
contain such element.\n *\n * The operation is _terminal_.\n */\npublic inline fun <T>
Sequence<T>.indexOfLast(predicate: (T) -> Boolean): Int {\n    var lastIndex = -1\n    var index = 0\n    for (item in
this) {\n        checkIndexOverflow(index)\n        if (predicate(item))\n            lastIndex = index\n            index++\n    }\n    return lastIndex\n}\n\n/**\n * Returns the last element.\n *\n * The operation is _terminal_.\n *\n * @throws
NoSuchElementException if the sequence is empty.\n *\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun <T> Sequence<T>.last(): T {\n    val iterator =
iterator()\n    if (!iterator.hasNext())\n        throw NoSuchElementException("Sequence is empty.")\n    var last =
iterator.next()\n    while (iterator.hasNext())\n        last = iterator.next()\n    return last\n}\n\n/**\n * Returns the last
element matching the given [predicate].\n *\n * The operation is _terminal_.\n *\n * @throws
NoSuchElementException if no such element is found.\n *\n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun <T> Sequence<T>.last(predicate: (T) ->
Boolean): T {\n    var last: T? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element))
{\n            last = element\n            found = true\n        }\n    }\n    if (!found) throw
NoSuchElementException("Sequence contains no element matching the predicate.")\n}\n\n@Suppress("UNCHECKED_CAST")\nreturn last as T\n}\n\n/**\n * Returns last index of [element], or -1 if the
sequence does not contain element.\n *\n * The operation is _terminal_.\n */\npublic fun
<@kotlin.internal.OnlyInputTypes T> Sequence<T>.lastIndexOf(element: T): Int {\n    var lastIndex = -1\n    var
index = 0\n    for (item in this) {\n        checkIndexOverflow(index)\n        if (element == item)\n            lastIndex =
index\n            index++\n    }\n    return lastIndex\n}\n\n/**\n * Returns the last element, or `null` if the sequence is

```

```

empty.\n *\n * The operation is _terminal_.\n *\n * @sample samples.collections.Collections.Elements.last\n
*\npublic fun <T> Sequence<T>.lastOrNull(): T? {\n    val iterator = iterator()\n    if (!iterator.hasNext())\n    return null\n    var last = iterator.next()\n    while (iterator.hasNext())\n        last = iterator.next()\n    return\n    last}\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n
*\n * The operation is _terminal_.\n *\n * @sample samples.collections.Collections.Elements.last\n
*\npublic\ninline fun <T> Sequence<T>.lastOrNull(predicate: (T) -> Boolean): T? {\n    var last: T? = null\n    for (element in\n    this) {\n        if (predicate(element)) {\n            last = element\n        }\n    }\n    return last}\n}\n\n/**\n * Returns the\nsingle element, or throws an exception if the sequence is empty or has more than one element.\n
*\n * The operation\nis _terminal_.\n
*\npublic fun <T> Sequence<T>.single(): T {\n    val iterator = iterator()\n    if\n    (!iterator.hasNext())\n        throw NoSuchElementException("Sequence is empty.")\n    val single =\n    iterator.next()\n    if (iterator.hasNext())\n        throw IllegalArgumentException("Sequence has more than one\n    element.")\n    return single}\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws\nexception if there is no or more than one matching element.\n
*\n * The operation is _terminal_.\n
*\npublic inline\nfun <T> Sequence<T>.single(predicate: (T) -> Boolean): T {\n    var single: T? = null\n    var found = false\n    for\n    (element in this) {\n        if (predicate(element)) {\n            if (found) throw IllegalArgumentException("Sequence\n    contains more than one matching element.")\n            single = element\n            found = true\n        }\n    }\n    if\n    (!found) throw NoSuchElementException("Sequence contains no element matching the predicate.")\n    @Suppress("UNCHECKED_CAST")\n    return single as T}\n}\n\n/**\n * Returns single element, or `null` if the\nsequence is empty or has more than one element.\n
*\n * The operation is _terminal_.\n
*\npublic fun <T>\nSequence<T>.singleOrNull(): T? {\n    val iterator = iterator()\n    if (!iterator.hasNext())\n        return null\n    val\n    single = iterator.next()\n    if (iterator.hasNext())\n        return null\n    return single}\n}\n\n/**\n * Returns the single\nelement matching the given [predicate], or `null` if element was not found or more than one element was found.\n
*\n * The operation is _terminal_.\n
*\npublic inline fun <T> Sequence<T>.singleOrNull(predicate: (T) ->\nBoolean): T? {\n    var single: T? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element))\n        {\n            if (found) return null\n            single = element\n            found = true\n        }\n    }\n    if\n    (!found) return\n    null\n    return single}\n}\n\n/**\n * Returns a sequence containing all elements except first [n] elements.\n
*\n * The\noperation is _intermediate_ and _stateless_.\n
*\n * @throws IllegalArgumentException if [n] is negative.\n
*\n * @sample samples.collections.Collections.Transformations.drop\n
*\npublic fun <T> Sequence<T>.drop(n: Int):\nSequence<T> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    return when {\n        n\n        == 0 -> this\n        this is DropTakeSequence -> this.drop(n)\n        else -> DropSequence(this, n)\n    }\n}\n\n/**\n * Returns a sequence containing all elements except first elements that satisfy the given [predicate].\n
*\n * The\noperation is _intermediate_ and _stateless_.\n
*\n * @sample\nsamples.collections.Collections.Transformations.drop\n
*\npublic fun <T> Sequence<T>.dropWhile(predicate: (T)\n-> Boolean): Sequence<T> {\n    return DropWhileSequence(this, predicate)\n}\n\n/**\n * Returns a sequence\ncontaining only elements matching the given [predicate].\n
*\n * The operation is _intermediate_ and _stateless_.\n
*\n * @sample samples.collections.Collections.Filtering.filter\n
*\npublic fun <T> Sequence<T>.filter(predicate:\n(T) -> Boolean): Sequence<T> {\n    return FilteringSequence(this, true, predicate)\n}\n\n/**\n * Returns a sequence\ncontaining only elements matching the given [predicate].\n
*\n * @param [predicate] function that takes the index of an\n    element and the element itself\n    * and returns the result of predicate evaluation on the element.\n
*\n * The\noperation is _intermediate_ and _stateless_.\n
*\n * @sample\nsamples.collections.Collections.Filtering.filterIndexed\n
*\npublic fun <T> Sequence<T>.filterIndexed(predicate:\n(index: Int, T) -> Boolean): Sequence<T> {\n    // TODO: Rewrite with generalized MapFilterIndexingSequence\n    return TransformingSequence(FilteringSequence(IndexingSequence(this), true, { predicate(it.index, it.value) }), {\n    it.value })\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n
*\n * @param\n[predicate] function that takes the index of an element and the element itself\n    * and returns the result of predicate\n    evaluation on the element.\n
*\n * The operation is _terminal_.\n
*\n * @sample\nsamples.collections.Collections.Filtering.filterIndexedTo\n
*\npublic inline fun <T, C : MutableCollection<in T>>\nSequence<T>.filterIndexedTo(destination: C, predicate: (index: Int, T) -> Boolean): C {\n    forEachIndexed {

```

```

index, element ->\n    if (predicate(index, element)) destination.add(element)\n } \n return
destination\n}\n\n/**\n * Returns a sequence containing all elements that are instances of specified type parameter
R.\n *\n * The operation is _intermediate_ and _stateless_.\n *\n * @sample
samples.collections.Collections.Filtering.filterIsInstance\n *\npublic inline fun <reified R>
Sequence<*>.filterIsInstance(): Sequence<@kotlin.internal.NoInfer R> {\n
@Suppress("UNCHECKED_CAST")\n    return filter { it is R } as Sequence<R>\n}\n\n/**\n * Appends all
elements that are instances of specified type parameter R to the given [destination].\n *\n * The operation is
_terminal_.\n *\n * @sample samples.collections.Collections.Filtering.filterIsInstanceTo\n *\npublic inline fun
<reified R, C : MutableCollection<in R>> Sequence<*>.filterIsInstanceTo(destination: C): C {\n    for (element in
this) if (element is R) destination.add(element)\n    return destination\n}\n\n/**\n * Returns a sequence containing
all elements not matching the given [predicate].\n *\n * The operation is _intermediate_ and _stateless_.\n *\n *
@sample samples.collections.Collections.Filtering.filter\n *\npublic fun <T> Sequence<T>.filterNot(predicate: (T)
-> Boolean): Sequence<T> {\n    return FilteringSequence(this, false, predicate)\n}\n\n/**\n * Returns a sequence
containing all elements that are not `null`.\n *\n * The operation is _intermediate_ and _stateless_.\n *\n * @sample
samples.collections.Collections.Filtering.filterNotNull\n *\npublic fun <T : Any> Sequence<T?>.filterNotNull():
Sequence<T> {\n    @Suppress("UNCHECKED_CAST")\n    return filterNot { it == null } as
Sequence<T>\n}\n\n/**\n * Appends all elements that are not `null` to the given [destination].\n *\n * The operation
is _terminal_.\n *\n * @sample samples.collections.Collections.Filtering.filterNotNullTo\n *\npublic fun <C :
MutableCollection<in T>, T : Any> Sequence<T?>.filterNotNullTo(destination: C): C {\n    for (element in this) if
(element != null) destination.add(element)\n    return destination\n}\n\n/**\n * Appends all elements not matching
the given [predicate] to the given [destination].\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Filtering.filterTo\n *\npublic inline fun <T, C : MutableCollection<in T>>
Sequence<T>.filterNotTo(destination: C, predicate: (T) -> Boolean): C {\n    for (element in this) if
(!predicate(element)) destination.add(element)\n    return destination\n}\n\n/**\n * Appends all elements matching
the given [predicate] to the given [destination].\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Filtering.filterTo\n *\npublic inline fun <T, C : MutableCollection<in T>>
Sequence<T>.filterTo(destination: C, predicate: (T) -> Boolean): C {\n    for (element in this) if (predicate(element))
destination.add(element)\n    return destination\n}\n\n/**\n * Returns a sequence containing first [n] elements.\n *\n *
The operation is _intermediate_ and _stateless_.\n *\n * @throws IllegalArgumentException if [n] is negative.\n *\n *
@sample samples.collections.Collections.Transformations.take\n *\npublic fun <T> Sequence<T>.take(n: Int):
Sequence<T> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    return when {\n        n
== 0 -> emptySequence()\n        this is DropTakeSequence -> this.take(n)\n        else -> TakeSequence(this, n)\n
}\n}\n\n/**\n * Returns a sequence containing first elements satisfying the given [predicate].\n *\n * The operation
is _intermediate_ and _stateless_.\n *\n * @sample samples.collections.Collections.Transformations.take\n *\npublic fun <T> Sequence<T>.takeWhile(predicate: (T) -> Boolean): Sequence<T> {\n    return
TakeWhileSequence(this, predicate)\n}\n\n/**\n * Returns a sequence that yields elements of this sequence sorted
according to their natural sort order.\n *\n * The sort is _stable_. It means that equal elements preserve their order
relative to each other after sorting.\n *\n * The operation is _intermediate_ and _stateful_.\n *\npublic fun <T :
Comparable<T>> Sequence<T>.sorted(): Sequence<T> {\n    return object : Sequence<T> {\n        override fun
iterator(): Iterator<T> {\n            val sortedList = this@sorted.toMutableList()\n            sortedList.sort()\n
return sortedList.iterator()\n        }\n    }\n}\n\n/**\n * Returns a sequence that yields elements of this sequence
sorted according to natural sort order of the value returned by specified [selector] function.\n *\n * The sort is
_stable_. It means that equal elements preserve their order relative to each other after sorting.\n *\n * The operation
is _intermediate_ and _stateful_.\n *\n * @sample samples.collections.Collections.Sorting.sortedBy\n *\npublic
inline fun <T, R : Comparable<R>> Sequence<T>.sortedBy(crossinline selector: (T) -> R?): Sequence<T> {\n
return sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a sequence that yields elements of this sequence
sorted descending according to natural sort order of the value returned by specified [selector] function.\n *\n * The
sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n *\n * The

```

```

operation is _intermediate_ and _stateful_.\n *\npublic inline fun <T, R : Comparable<R>>
Sequence<T>.sortedByDescending(crossinline selector: (T) -> R?): Sequence<T> {\n  return
sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a sequence that yields elements of this sequence
sorted descending according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements
preserve their order relative to each other after sorting.\n * \n * The operation is _intermediate_ and _stateful_.\n
*\npublic fun <T : Comparable<T>> Sequence<T>.sortedDescending(): Sequence<T> {\n  return
sortedWith(reverseOrder())\n}\n\n/**\n * Returns a sequence that yields elements of this sequence sorted according
to the specified [comparator].\n * \n * The sort is _stable_. It means that equal elements preserve their order relative
to each other after sorting.\n * \n * The operation is _intermediate_ and _stateful_.\n *\npublic fun <T>
Sequence<T>.sortedWith(comparator: Comparator<in T>): Sequence<T> {\n  return object : Sequence<T> {\n
override fun iterator(): Iterator<T> {\n      val sortedList = this@sortedWith.toMutableList()\n
sortedList.sortWith(comparator)\n      return sortedList.iterator()\n    }\n  }\n}\n\n/**\n * Returns a [Map]
containing key-value pairs provided by [transform] function\n * applied to elements of the given sequence.\n * \n *
If any of two pairs would have the same key the last one gets added to the map.\n * \n * The returned map preserves
the entry iteration order of the original sequence.\n * \n * The operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Transformations.associate\n *\npublic inline fun <T, K, V>
Sequence<T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> {\n  return associateTo(LinkedHashMap<K,
V>(), transform)\n}\n\n/**\n * Returns a [Map] containing the elements from the given sequence indexed by the
key\n * returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the
same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry
iteration order of the original sequence.\n * \n * The operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Transformations.associateBy\n *\npublic inline fun <T, K>
Sequence<T>.associateBy(keySelector: (T) -> K): Map<K, T> {\n  return associateByTo(LinkedHashMap<K,
T>(), keySelector)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform] and indexed
by [keySelector] functions applied to elements of the given sequence.\n * \n * If any two elements would have the
same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry
iteration order of the original sequence.\n * \n * The operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Transformations.associateByWithValueTransform\n *\npublic inline fun <T, K, V>
Sequence<T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> {\n  return
associateByTo(LinkedHashMap<K, V>(), keySelector, valueTransform)\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to
each element of the given sequence\n * and value is the element itself.\n * \n * If any two elements would have the
same key returned by [keySelector] the last one gets added to the map.\n * \n * The operation is _terminal_.\n * \n *
@sample samples.collections.Collections.Transformations.associateByTo\n *\npublic inline fun <T, K, M :
MutableMap<in K, in T>> Sequence<T>.associateByTo(destination: M, keySelector: (T) -> K): M {\n  for
(element in this) {\n      destination.put(keySelector(element), element)\n    }\n  }\n  return destination\n}\n\n/**\n *
Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the
[keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the
given sequence.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets
added to the map.\n * \n * The operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Transformations.associateByToWithValueTransform\n *\npublic inline fun <T, K,
V, M : MutableMap<in K, in V>> Sequence<T>.associateByTo(destination: M, keySelector: (T) -> K,
valueTransform: (T) -> V): M {\n  for (element in this) {\n      destination.put(keySelector(element),
valueTransform(element))\n    }\n  }\n  return destination\n}\n\n/**\n * Populates and returns the [destination] mutable
map with key-value pairs\n * provided by [transform] function applied to each element of the given sequence.\n * \n *
If any of two pairs would have the same key the last one gets added to the map.\n * \n * The operation is
_terminal_.\n * \n * @sample samples.collections.Collections.Transformations.associateTo\n *\npublic inline fun
<T, K, V, M : MutableMap<in K, in V>> Sequence<T>.associateTo(destination: M, transform: (T) -> Pair<K, V>):

```

```

M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

Returns a [Map] where keys are elements from the given sequence and values are produced by the [valueSelector] function applied to each element.
If any two elements are equal, the last one gets added to the map.
The returned map preserves the entry iteration order of the original sequence.
The operation is _terminal_.

@sample samples.collections.Collections.Transformations.associateWith

Since Kotlin("1.3")
public inline fun <K, V> Sequence<K>.associateWith(valueSelector: (K) -> V): Map<K, V> {
    val result = LinkedHashMap<K, V>()
    return associateWithTo(result, valueSelector)
}

Populates and returns the [destination] mutable map with key-value pairs for each element of the given sequence, where key is the element itself and value is provided by the [valueSelector] function applied to that key.
If any two elements are equal, the last one overwrites the former value in the map.
The operation is _terminal_.

@sample samples.collections.Collections.Transformations.associateWithTo

Since Kotlin("1.3")
public inline fun <K, V, M : MutableMap<in K, in V>> Sequence<K>.associateWithTo(destination: M, valueSelector: (K) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

Appends all elements to the given [destination] collection.
The operation is _terminal_.

public fun <T, C : MutableCollection<in T>> Sequence<T>.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

Returns a new [HashSet] of all elements.
The operation is _terminal_.

public fun <T> Sequence<T>.toHashSet(): HashSet<T> {
    return toCollection(HashSet<T>())
}

Returns a [List] containing all elements.
The operation is _terminal_.

public fun <T> Sequence<T>.toList(): List<T> {
    return this.toMutableList().optimizeReadOnlyList()
}

Returns a new [MutableList] filled with all elements of this sequence.
The operation is _terminal_.

public fun <T> Sequence<T>.toMutableList(): MutableList<T> {
    return toCollection(ArrayList<T>())
}

Returns a [Set] of all elements.
The returned set preserves the element iteration order of the original sequence.
The operation is _terminal_.

public fun <T> Sequence<T>.toSet(): Set<T> {
    return toCollection(LinkedHashSet<T>()).optimizeReadOnlySet()
}

Returns a single sequence of all elements from results of [transform] function being invoked on each element of original sequence.
The operation is _intermediate_ and _stateless_.

@sample samples.collections.Collections.Transformations.flatMap

Since Kotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIterable")
public fun <T, R> Sequence<T>.flatMap(transform: (T) -> Iterable<R>): Sequence<R> {
    return FlatteningSequence(this, transform, Iterable<R>::iterator)
}

Returns a single sequence of all elements from results of [transform] function being invoked on each element of original sequence.
The operation is _intermediate_ and _stateless_.

@sample samples.collections.Collections.Transformations.flatMap

public fun <T, R> Sequence<T>.flatMap(transform: (T) -> Sequence<R>): Sequence<R> {
    return FlatteningSequence(this, transform, Sequence<R>::iterator)
}

Returns a single sequence of all elements yielded from results of [transform] function being invoked on each element and its index in the original sequence.
The operation is _intermediate_ and _stateless_.

@sample samples.collections.Collections.Transformations.flatMapIndexed

Since Kotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterable")
public fun <T, R> Sequence<T>.flatMapIndexed(transform: (index: Int, T) -> Iterable<R>): Sequence<R> {
    return flatMapIndexed(this, transform, Iterable<R>::iterator)
}

Returns a single sequence of all elements yielded from results of [transform] function being invoked on each element and its index in the original sequence.
The operation is _intermediate_ and _stateless_.

@sample samples.collections.Collections.Transformations.flatMapIndexed

Since Kotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedSequence")
public fun <T, R>

```

```

Sequence<T>.flatMapIndexed(transform: (index: Int, T) -> Sequence<R>): Sequence<R> {\n  return
flatMapIndexed(this, transform, Sequence<R>::iterator)\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original sequence, to the given
[destination].\n * \n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <T, R, C : MutableCollection<in R>> Sequence<T>.flatMapIndexedTo(destination: C, transform:
(index: Int, T) -> Iterable<R>): C {\n  var index = 0\n  for (element in this) {\n    val list =
transform(checkIndexOverflow(index++), element)\n    destination.addAll(list)\n  }\n  return
destination\n}\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element\n * and its index in the original sequence, to the given [destination].\n * \n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapIndexedSequenceTo")\n@kotlin.internal.InlineOnly\npu
blic inline fun <T, R, C : MutableCollection<in R>> Sequence<T>.flatMapIndexedTo(destination: C, transform:
(index: Int, T) -> Sequence<R>): C {\n  var index = 0\n  for (element in this) {\n    val list =
transform(checkIndexOverflow(index++), element)\n    destination.addAll(list)\n  }\n  return
destination\n}\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original sequence, to the given [destination].\n * \n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapIterableTo")\n\npublic inline fun <T, R, C :
MutableCollection<in R>> Sequence<T>.flatMapTo(destination: C, transform: (T) -> Iterable<R>): C {\n  for
(element in this) {\n    val list = transform(element)\n    destination.addAll(list)\n  }\n  return
destination\n}\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original sequence, to the given [destination].\n * \n * The operation is _terminal_.\n
*\n@public inline fun
<T, R, C : MutableCollection<in R>> Sequence<T>.flatMapTo(destination: C, transform: (T) -> Sequence<R>): C
{\n  for (element in this) {\n    val list = transform(element)\n    destination.addAll(list)\n  }\n  return
destination\n}\n}\n\n/**\n * Groups elements of the original sequence by the key returned by the given [keySelector]
function\n * applied to each element and returns a map where each group key is associated with a list of
corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the
original sequence.\n * \n * The operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n *\n@public inline fun <T, K>
Sequence<T>.groupBy(keySelector: (T) -> K): Map<K, List<T>> {\n  return groupByTo(LinkedHashMap<K,
MutableList<T>>(), keySelector)\n}\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to
each element of the original sequence\n * by the key returned by the given [keySelector] function applied to the
element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The
returned map preserves the entry iteration order of the keys produced from the original sequence.\n * \n * The
operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n *\n@public inline fun <T, K, V>
Sequence<T>.groupBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, List<V>> {\n  return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n}\n\n/**\n * Groups elements
of the original sequence by the key returned by the given [keySelector] function\n * applied to each element and
puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The
[destination] map.\n * \n * The operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n *\n@public inline fun <T, K, M : MutableMap<in K,
MutableList<T>>> Sequence<T>.groupByTo(destination: M, keySelector: (T) -> K): M {\n  for (element in this)
{\n    val key = keySelector(element)\n    val list = destination.getOrPut(key) { ArrayList<T>() }\n
list.add(element)\n  }\n  return destination\n}\n}\n\n/**\n * Groups values returned by the [valueTransform] function
applied to each element of the original sequence\n * by the key returned by the given [keySelector] function applied

```



```

destination.add(transform(item))\n return destination\n}\n\n/**\n * Returns a sequence that wraps each element of
the original sequence\n * into an [IndexedValue] containing the index of that element and the element itself.\n *\n * The operation is _intermediate_ and _stateless_.\n */\npublic fun <T> Sequence<T>.withIndex():
Sequence<IndexedValue<T>> {\n return IndexingSequence(this)\n}\n\n/**\n * Returns a sequence containing
only distinct elements from the given sequence.\n *\n * Among equal elements of the given sequence, only the first
one will be present in the resulting sequence.\n *\n * The elements in the resulting sequence are in the same order as
they were in the source sequence.\n *\n * The operation is _intermediate_ and _stateful_.\n */\n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic fun <T>
Sequence<T>.distinct(): Sequence<T> {\n return this.distinctBy { it }\n}\n\n/**\n * Returns a sequence
containing only elements from the given sequence\n * having distinct keys returned by the given [selector]
function.\n *\n * Among elements of the given sequence with equal keys, only the first one will be present in the
resulting sequence.\n *\n * The elements in the resulting sequence are in the same order as they were in the source
sequence.\n *\n * The operation is _intermediate_ and _stateful_.\n */\n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic fun <T, K>
Sequence<T>.distinctBy(selector: (T) -> K): Sequence<T> {\n return DistinctSequence(this, selector)\n}\n\n/**\n *
Returns a new [MutableSet] containing all distinct elements from the given sequence.\n *\n * The returned set
preserves the element iteration order of the original sequence.\n *\n * The operation is _terminal_.\n */\npublic fun
<T> Sequence<T>.toMutableSet(): MutableSet<T> {\n val set = LinkedHashSet<T>()\n for (item in this)
set.add(item)\n return set\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n *\n * The
operation is _terminal_.\n */\n * @sample samples.collections.Collections.Aggregates.all\n */\npublic inline fun
<T> Sequence<T>.all(predicate: (T) -> Boolean): Boolean {\n for (element in this) if (!predicate(element)) return
false\n return true\n}\n\n/**\n * Returns `true` if sequence has at least one element.\n *\n * The operation is
_terminal_.\n */\n * @sample samples.collections.Collections.Aggregates.any\n */\npublic inline fun <T>
Sequence<T>.any(): Boolean {\n return iterator().hasNext()\n}\n\n/**\n * Returns `true` if at least one element
matches the given [predicate].\n *\n * The operation is _terminal_.\n */\n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n */\npublic inline fun <T>
Sequence<T>.any(predicate: (T) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return
true\n return false\n}\n\n/**\n * Returns the number of elements in this sequence.\n *\n * The operation is
_terminal_.\n */\npublic fun <T> Sequence<T>.count(): Int {\n var count = 0\n for (element in this)
checkCountOverflow(++count)\n return count\n}\n\n/**\n * Returns the number of elements matching the given
[predicate].\n *\n * The operation is _terminal_.\n */\npublic inline fun <T> Sequence<T>.count(predicate: (T) ->
Boolean): Int {\n var count = 0\n for (element in this) if (predicate(element)) checkCountOverflow(++count)\n
return count\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to
right\n * to current accumulator value and each element.\n *\n * Returns the specified [initial] value if the sequence
is empty.\n *\n * @param [operation] function that takes current accumulator value and an element, and calculates
the next accumulator value.\n *\n * The operation is _terminal_.\n */\npublic inline fun <T, R>
Sequence<T>.fold(initial: R, operation: (acc: R, T) -> R): R {\n var accumulator = initial\n for (element in this)
accumulator = operation(accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting
with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element
with its index in the original sequence.\n *\n * Returns the specified [initial] value if the sequence is empty.\n *\n *
@param [operation] function that takes the index of an element, current accumulator value\n * and the element
itself, and calculates the next accumulator value.\n *\n * The operation is _terminal_.\n */\npublic inline fun <T, R>
Sequence<T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {\n var index = 0\n var
accumulator = initial\n for (element in this) accumulator = operation(checkIndexOverflow(index++), accumulator,
element)\n return accumulator\n}\n\n/**\n * Performs the given [action] on each element.\n *\n * The operation is
_terminal_.\n */\npublic inline fun <T> Sequence<T>.forEach(action: (T) -> Unit): Unit {\n for (element in this)
action(element)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the
element.\n *\n * @param [action] function that takes the index of an element and the element itself\n * and performs the

```



```

action on the element.\n * \n * The operation is _terminal_.\n * \n\npublic inline fun <T>
Sequence<T>.forEachIndexed(action: (index: Int, T) -> Unit): Unit {\n    var index = 0\n    for (item in this)
action(checkIndexOverflow(index++), item)\n}\n\n/**\n * Returns the largest element.\n * \n * If any of elements is
`NaN` returns `NaN`.\n * \n * The operation is _terminal_.\n * \n * @throws NoSuchElementException if the
sequence is empty.\n
*\n*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
Sequence<Double>.max(): Double {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw
NoSuchElementException()\n    var max = iterator.next()\n    while (iterator.hasNext()) {\n        val e =
iterator.next()\n        max = maxOf(max, e)\n    }\n    return max\n}\n\n/**\n * Returns the largest element.\n * \n *
If any of elements is `NaN` returns `NaN`.\n * \n * The operation is _terminal_.\n * \n * @throws
NoSuchElementException if the sequence is empty.\n
*\n*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
Sequence<Float>.max(): Float {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw
NoSuchElementException()\n    var max = iterator.next()\n    while (iterator.hasNext()) {\n        val e =
iterator.next()\n        max = maxOf(max, e)\n    }\n    return max\n}\n\n/**\n * Returns the largest element.\n * \n *
The operation is _terminal_.\n * \n * @throws NoSuchElementException if the sequence is empty.\n
*\n*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
<T : Comparable<T>> Sequence<T>.max(): T {\n    val iterator = iterator()\n    if
(iterator.hasNext()) throw NoSuchElementException()\n    var max = iterator.next()\n    while (iterator.hasNext())
{\n        val e = iterator.next()\n        if (max < e) max = e\n    }\n    return max\n}\n\n/**\n * Returns the first
element yielding the largest value of the given function.\n * \n * The operation is _terminal_.\n * \n * @throws
NoSuchElementException if the sequence is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.maxBy\n
*\n*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic inline fun
<T, R : Comparable<R>> Sequence<T>.maxBy(selector: (T) -> R): T {\n    val iterator
= iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var maxElem = iterator.next()\n    if
(iterator.hasNext()) return maxElem\n    var maxValue = selector(maxElem)\n    do {\n        val e = iterator.next()\n
        val v = selector(e)\n        if (maxValue < v) {\n            maxElem = e\n            maxValue = v\n        }\n    } while
(iterator.hasNext())\n    return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the
given function or `null` if there are no elements.\n * \n * The operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Aggregates.maxByOrNull\n
*\n*\n@SinceKotlin("1.4")\npublic inline fun <T, R :
Comparable<R>> Sequence<T>.maxByOrNull(selector: (T) -> R): T? {\n    val iterator = iterator()\n    if
(iterator.hasNext()) return null\n    var maxElem = iterator.next()\n    if (!iterator.hasNext()) return maxElem\n    var
maxValue = selector(maxElem)\n    do {\n        val e = iterator.next()\n        val v = selector(e)\n        if (maxValue <
v) {\n            maxElem = e\n            maxValue = v\n        }\n    } while (iterator.hasNext())\n    return
maxElem\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the sequence.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result
is `NaN`.\n * \n * The operation is _terminal_.\n * \n * @throws NoSuchElementException if the sequence is
empty.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.maxOf(selector: (T) ->
Double): Double {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var
maxValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n
        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all
values produced by [selector] function\n * applied to each element in the sequence.\n * \n * If any of values
produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * The operation is _terminal_.\n * \n *
@throws NoSuchElementException if the sequence is empty.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

```

```

ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.maxOf(selector: (T) ->
Float): Float {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var
maxValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n
maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all
values produced by [selector] function\n * applied to each element in the sequence.\n * The operation is
_terminal_.\n * @throws NoSuchElementException if the sequence is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>
Sequence<T>.maxOf(selector: (T) -> R): R {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw
NoSuchElementException()\n    var maxValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v
= selector(iterator.next())\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the sequence or `null` if there are no elements.\n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.maxOfOrNull(selector:
(T) -> Double): Double? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var
maxValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n
maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the sequence or `null` if there are no elements.\n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.maxOfOrNull(selector:
(T) -> Float): Float? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var
maxValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n
maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the sequence or `null` if there are no elements.\n * The
operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>
Sequence<T>.maxOfOrNull(selector: (T) -> R): R? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return
null\n    var maxValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v =
selector(iterator.next())\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return
maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the sequence.\n * @throws
NoSuchElementException if the sequence is empty.\n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>
Sequence<T>.maxOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n    val iterator = iterator()\n    if
(!iterator.hasNext()) throw NoSuchElementException()\n    var maxValue = selector(iterator.next())\n    while
(iterator.hasNext()) {\n        val v = selector(iterator.next())\n        if (comparator.compare(maxValue, v) < 0) {\n
            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each element in the sequence or `null`
if there are no elements.\n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>
Sequence<T>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n    val iterator =
iterator()\n    if (!iterator.hasNext()) return null\n    var maxValue = selector(iterator.next())\n    while

```



```

(iterator.hasNext()) return minElem\n    var minValue = selector(minElem)\n    do {\n        val e = iterator.next()\n        val v = selector(e)\n        if (minValue > v) {\n            minElem = e\n            minValue = v\n        }\n    } while
(iterator.hasNext())\n    return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the
given function or `null` if there are no elements.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Aggregates.minByOrNull\n *\n * @SinceKotlin("1.4")\n\npublic inline fun <T, R :
Comparable<R>> Sequence<T>.minByOrNull(selector: (T) -> R): T? {\n    val iterator = iterator()\n    if
(iterator.hasNext()) return null\n    var minElem = iterator.next()\n    if (!iterator.hasNext()) return minElem\n    var
minValue = selector(minElem)\n    do {\n        val e = iterator.next()\n        val v = selector(e)\n        if (minValue >
v) {\n            minElem = e\n            minValue = v\n        }\n    } while (iterator.hasNext())\n    return
minElem\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the sequence.\n *\n * If any of values produced by [selector] function is `NaN`, the returned result
is `NaN`.\n *\n * The operation is _terminal_.\n *\n * @throws NoSuchElementException if the sequence is
empty.\n
*\n * @SinceKotlin("1.4")\n * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n * @OverloadResolution
ByLambdaReturnType\n * @kotlin.internal.InlineOnly\n\npublic inline fun <T> Sequence<T>.minOf(selector: (T) ->
Double): Double {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var
minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n
minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all
values produced by [selector] function\n * applied to each element in the sequence.\n *\n * If any of values
produced by [selector] function is `NaN`, the returned result is `NaN`.\n *\n * The operation is _terminal_.\n *\n *
@throws NoSuchElementException if the sequence is empty.\n
*\n * @SinceKotlin("1.4")\n * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n * @OverloadResolution
ByLambdaReturnType\n * @kotlin.internal.InlineOnly\n\npublic inline fun <T> Sequence<T>.minOf(selector: (T) ->
Float): Float {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var
minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n
minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all
values produced by [selector] function\n * applied to each element in the sequence.\n *\n * The operation is
_terminal_.\n *\n * @throws NoSuchElementException if the sequence is empty.\n
*\n * @SinceKotlin("1.4")\n * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n * @OverloadResolution
ByLambdaReturnType\n * @kotlin.internal.InlineOnly\n\npublic inline fun <T, R : Comparable<R>>
Sequence<T>.minOf(selector: (T) -> R): R {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw
NoSuchElementException()\n    var minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v
= selector(iterator.next())\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the sequence or `null` if there are no elements.\n *\n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n *\n * The operation is _terminal_.\n
*\n * @SinceKotlin("1.4")\n * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n * @OverloadResolution
ByLambdaReturnType\n * @kotlin.internal.InlineOnly\n\npublic inline fun <T> Sequence<T>.minOfOrNull(selector:
(T) -> Double): Double? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var minValue =
selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        minValue =
minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the sequence or `null` if there are no elements.\n *\n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n *\n * The operation is _terminal_.\n
*\n * @SinceKotlin("1.4")\n * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n * @OverloadResolution
ByLambdaReturnType\n * @kotlin.internal.InlineOnly\n\npublic inline fun <T> Sequence<T>.minOfOrNull(selector:
(T) -> Float): Float? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var minValue =
selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        minValue =
minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced

```

by [selector] function\n * applied to each element in the sequence or `null` if there are no elements.\n *\n * The operation is _terminal_.\n

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>
Sequence<T>.minOrNull(selector: (T) -> R): R? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return
null\n    var minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v =
selector(iterator.next())\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return
minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the sequence.\n * \n * @throws
NoSuchElementException if the sequence is empty.\n *\n * The operation is _terminal_.\n
```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>
Sequence<T>.minOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n    val iterator = iterator()\n    if
(!iterator.hasNext()) throw NoSuchElementException()\n    var minValue = selector(iterator.next())\n    while
(iterator.hasNext()) {\n        val v = selector(iterator.next())\n        if (comparator.compare(minValue, v) > 0) {\n
            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value according to the
provided [comparator]\n * among all values produced by [selector] function applied to each element in the sequence
or `null` if there are no elements.\n *\n * The operation is _terminal_.\n
```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>
Sequence<T>.minOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n    val iterator =
iterator()\n    if (!iterator.hasNext()) return null\n    var minValue = selector(iterator.next())\n    while
(iterator.hasNext()) {\n        val v = selector(iterator.next())\n        if (comparator.compare(minValue, v) > 0) {\n
            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest element or `null` if there are
no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n *\n * The operation is _terminal_.\n
```

```

*\n@SinceKotlin("1.4")\npublic fun Sequence<Double>.minOrNull(): Double? {\n    val iterator = iterator()\n    if
(!iterator.hasNext()) return null\n    var min = iterator.next()\n    while (iterator.hasNext()) {\n        val e =
iterator.next()\n        min = minOf(min, e)\n    }\n    return min\n}\n\n/**\n * Returns the smallest element or `null` if
there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n *\n * The operation is _terminal_.\n
```

```

*\n@SinceKotlin("1.4")\npublic fun Sequence<Float>.minOrNull(): Float? {\n    val iterator = iterator()\n    if
(!iterator.hasNext()) return null\n    var min = iterator.next()\n    while (iterator.hasNext()) {\n        val e =
iterator.next()\n        min = minOf(min, e)\n    }\n    return min\n}\n\n/**\n * Returns the smallest element or `null` if
there are no elements.\n *\n * The operation is _terminal_.\n
```

```

*\n@SinceKotlin("1.4")\npublic fun <T :
Comparable<T>> Sequence<T>.minOrNull(): T? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return
null\n    var min = iterator.next()\n    while (iterator.hasNext()) {\n        val e = iterator.next()\n        if (min > e) min
= e\n    }\n    return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided
[comparator].\n *\n * The operation is _terminal_.\n * \n * @throws NoSuchElementException if the sequence is
empty.\n
```

```

*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVER
LOADS")\npublic fun <T> Sequence<T>.minWith(comparator: Comparator<in T>): T {\n    val iterator =
iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var min = iterator.next()\n    while
(iterator.hasNext()) {\n        val e = iterator.next()\n        if (comparator.compare(min, e) > 0) min = e\n    }\n    return
min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or
`null` if there are no elements.\n *\n * The operation is _terminal_.\n
```

```

*\n@SinceKotlin("1.4")\npublic fun <T>
Sequence<T>.minWithOrNull(comparator: Comparator<in T>): T? {\n    val iterator = iterator()\n    if
(!iterator.hasNext()) return null\n    var min = iterator.next()\n    while (iterator.hasNext()) {\n        val e =
iterator.next()\n        if (comparator.compare(min, e) > 0) min = e\n    }\n    return min\n}\n\n/**\n * Returns `true` if
the sequence has no elements.\n *\n * The operation is _terminal_.\n * \n * @sample

```

```

samples.collections.Collections.Aggregates.none\n *\npublic fun <T> Sequence<T>.none(): Boolean {\n    return
!iterator().hasNext()\n}\n\n**\n * Returns `true` if no elements match the given [predicate].\n *\n * The operation is
_terminal_.\n *\n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n *\npublic inline fun
<T> Sequence<T>.none(predicate: (T) -> Boolean): Boolean {\n    for (element in this) if (predicate(element))
return false\n    return true\n}\n\n**\n * Returns a sequence which performs the given [action] on each element of
the original sequence as they pass through it.\n *\n * The operation is _intermediate_ and _stateless_.\n
*\n*@SinceKotlin("1.1")\npublic fun <T> Sequence<T>.onEach(action: (T) -> Unit): Sequence<T> {\n    return
map {\n        action(it)\n        it\n    }\n}\n\n**\n * Returns a sequence which performs the given [action] on each
element of the original sequence as they pass through it.\n *\n * @param [action] function that takes the index of an
element and the element itself\n * and performs the action on the element.\n *\n * The operation is _intermediate_
and _stateless_.\n *\n*@SinceKotlin("1.4")\npublic fun <T> Sequence<T>.onEachIndexed(action: (index: Int, T) -
> Unit): Sequence<T> {\n    return mapIndexed { index, element ->\n        action(index, element)\n        element\n
}\n}\n\n**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to
current accumulator value and each element.\n *\n * Throws an exception if this sequence is empty. If the sequence
can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is
empty.\n *\n * @param [operation] function that takes current accumulator value and an element,\n * and calculates
the next accumulator value.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Aggregates.reduce\n *\npublic inline fun <S, T : S>
Sequence<T>.reduce(operation: (acc: S, T) -> S): S {\n    val iterator = this.iterator()\n    if (!iterator.hasNext())
throw UnsupportedOperationException("Empty sequence can't be reduced.")\n    var accumulator: S =
iterator.next()\n    while (iterator.hasNext()) {\n        accumulator = operation(accumulator, iterator.next())\n    }\n
return accumulator\n}\n\n**\n * Accumulates value starting with the first element and applying [operation] from
left to right\n * to current accumulator value and each element with its index in the original sequence.\n *\n *
Throws an exception if this sequence is empty. If the sequence can be empty in an expected way,\n * please use
[reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n *\n * @param [operation] function
that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next
accumulator value.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Aggregates.reduce\n *\npublic inline fun <S, T : S>
Sequence<T>.reduceIndexed(operation: (index: Int, acc: S, T) -> S): S {\n    val iterator = this.iterator()\n    if
(!iterator.hasNext()) throw UnsupportedOperationException("Empty sequence can't be reduced.")\n    var index =
1\n    var accumulator: S = iterator.next()\n    while (iterator.hasNext()) {\n        accumulator =
operation(checkIndexOverflow(index++), accumulator, iterator.next())\n    }\n    return accumulator\n}\n\n**\n *
Accumulates value starting with the first element and applying [operation] from left to right\n * to current
accumulator value and each element with its index in the original sequence.\n *\n * Returns `null` if the sequence is
empty.\n *\n * @param [operation] function that takes the index of an element, current accumulator value and the
element itself,\n * and calculates the next accumulator value.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n *\n*@SinceKotlin("1.4")\npublic inline fun <S, T : S>
Sequence<T>.reduceIndexedOrNull(operation: (index: Int, acc: S, T) -> S): S? {\n    val iterator = this.iterator()\n
if (!iterator.hasNext()) return null\n    var index = 1\n    var accumulator: S = iterator.next()\n    while
(iterator.hasNext()) {\n        accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())\n
    }\n    return accumulator\n}\n\n**\n * Accumulates value starting with the first element and applying [operation]
from left to right\n * to current accumulator value and each element.\n *\n * Returns `null` if the sequence is
empty.\n *\n * @param [operation] function that takes current accumulator value and an element,\n * and calculates
the next accumulator value.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*\n*\n*@SinceKotlin("1.4")\n*@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S>
Sequence<T>.reduceOrNull(operation: (acc: S, T) -> S): S? {\n    val iterator = this.iterator()\n    if
(!iterator.hasNext()) return null\n    var accumulator: S = iterator.next()\n    while (iterator.hasNext()) {\n

```

```

accumulator = operation(accumulator, iterator.next())\n    }\n    return accumulator\n}\n\n/**\n * Returns a sequence
containing successive accumulation values generated by applying [operation] from left to right\n * to each element
and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation]
function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n * The
[initial] value should also be immutable (or should not be mutated)\n * as it may be passed to [operation] function
later because of sequence's lazy nature.\n * \n * @param [operation] function that takes current accumulator value
and an element, and calculates the next accumulator value.\n *\n * The operation is _intermediate_ and
_stateless_.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\npublic fun <T, R> Sequence<T>.runningFold(initial: R, operation: (acc: R, T) -> R):
Sequence<R> {\n    return sequence {\n        yield(initial)\n        var accumulator = initial\n        for (element in
this@runningFold) {\n            accumulator = operation(accumulator, element)\n            yield(accumulator)\n        }\n    }\n}\n\n/**\n * Returns a sequence containing successive accumulation values generated by applying [operation]
from left to right\n * to each element, its index in the original sequence and current accumulator value that starts
with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting sequence.\n * The [initial] value should also be immutable
(or should not be mutated)\n * as it may be passed to [operation] function later because of sequence's lazy nature.\n
*\n * @param [operation] function that takes the index of an element, current accumulator value\n * and the
element itself, and calculates the next accumulator value.\n *\n * The operation is _intermediate_ and
_stateless_.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\npublic fun
<T, R> Sequence<T>.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): Sequence<R> {\n
return sequence {\n    yield(initial)\n    var index = 0\n    var accumulator = initial\n    for (element in
this@runningFoldIndexed) {\n        accumulator = operation(checkIndexOverflow(index++), accumulator,
element)\n        yield(accumulator)\n    }\n}\n}\n\n/**\n * Returns a sequence containing successive
accumulation values generated by applying [operation] from left to right\n * to each element and current
accumulator value that starts with the first element of this sequence.\n * \n * Note that `acc` value passed to
[operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n
*\n * @param [operation] function that takes current accumulator value and the element, and calculates the next
accumulator value.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <S, T : S>
Sequence<T>.runningReduce(operation: (acc: S, T) -> S): Sequence<S> {\n    return sequence {\n        val iterator =
iterator()\n        if (iterator.hasNext()) {\n            var accumulator: S = iterator.next()\n            yield(accumulator)\n
            while (iterator.hasNext()) {\n                accumulator = operation(accumulator, iterator.next())\n
yield(accumulator)\n            }\n        }\n    }\n}\n\n/**\n * Returns a sequence containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original sequence and
current accumulator value that starts with the first element of this sequence.\n * \n * Note that `acc` value passed to
[operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n
*\n * @param [operation] function that takes the index of an element, current accumulator value\n * and the
element itself, and calculates the next accumulator value.\n *\n * The operation is _intermediate_ and
_stateless_.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n
*\n@SinceKotlin("1.4")\npublic fun
<S, T : S> Sequence<T>.runningReduceIndexed(operation: (index: Int, acc: S, T) -> S): Sequence<S> {\n    return
sequence {\n        val iterator = iterator()\n        if (iterator.hasNext()) {\n            var accumulator: S =
iterator.next()\n            yield(accumulator)\n            var index = 1\n            while (iterator.hasNext()) {\n
accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())\n
yield(accumulator)\n            }\n        }\n    }\n}\n\n/**\n * Returns a sequence containing successive accumulation
values generated by applying [operation] from left to right\n * to each element and current accumulator value that
starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting sequence.\n * The [initial] value should also be immutable

```

(or should not be mutated)\n * as it may be passed to [operation] function later because of sequence's lazy nature.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * The operation is `_intermediate_` and `_stateless_`.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n

```

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T, R>
Sequence<T>.scan(initial: R, operation: (acc: R, T) -> R): Sequence<R> {\n    return runningFold(initial,
operation)\n}\n\n/**\n * Returns a sequence containing successive accumulation values generated by applying
[operation] from left to right\n * to each element, its index in the original sequence and current accumulator value
that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n * The [initial] value should also be immutable
(or should not be mutated)\n * as it may be passed to [operation] function later because of sequence's lazy nature.\n
* \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the
element itself, and calculates the next accumulator value.\n *\n * The operation is _intermediate_ and _stateless_.\n
* \n * @sample samples.collections.Collections.Aggregates.scan\n

```

```

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T, R>
Sequence<T>.scanIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): Sequence<R> {\n    return
runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the sequence.\n * \n * The operation is _terminal_.\n *\n@Deprecated("Use sumOf
instead.", ReplaceWith("this.sumOf(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline
fun <T> Sequence<T>.sumBy(selector: (T) -> Int): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum
+= selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the sequence.\n * \n * The operation is _terminal_.\n *\n@Deprecated("Use
sumOf instead.", ReplaceWith("this.sumOf(selector)"))\n@DeprecatedSinceKotlin(warningSince =
"1.5")\npublic inline fun <T> Sequence<T>.sumByDouble(selector: (T) -> Double): Double {\n    var sum: Double
= 0.0\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum
of all values produced by [selector] function applied to each element in the sequence.\n * \n * The operation is
_terminal_.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
<T> Sequence<T>.sumOf(selector: (T) -> Double): Double {\n    var sum: Double = 0.toDouble()\n    for (element
in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced
by [selector] function applied to each element in the sequence.\n * \n * The operation is _terminal_.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Sequence<T>.sumOf(selector: (T) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum
+= selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the sequence.\n * \n * The operation is _terminal_.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun
<T> Sequence<T>.sumOf(selector: (T) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this)
{\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the sequence.\n * \n * The operation is _terminal_.\n

```

```

*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.sumOf(selector: (T) -> UInt): UInt {\n
var sum: UInt = 0.toUInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the
sequence.\n * \n * The operation is _terminal_.\n

```



```

*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.sumOf(selector: (T) -> ULong):
ULong {\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n**\n * Returns an original collection containing all the non-`null` elements, throwing an
[IllegalArgumentException] if there are any `null` elements.\n * \n * The operation is _intermediate_ and
_stateless_.\n *\npublic fun <T : Any> Sequence<T?>.requireNonNulls(): Sequence<T> {\n    return map { it ?\n        throw IllegalArgumentException("null element found in $this.") }\n}\n\n**\n * Splits this sequence into a
sequence of lists each not exceeding the given [size].\n * \n * The last list in the resulting sequence may have fewer
elements than the given [size].\n * \n * @param size the number of elements to take in each list, must be positive
and can be greater than the number of elements in this sequence.\n * \n * The operation is _intermediate_ and
_stateful_.\n * \n * @sample samples.collections.Collections.Transformations.chunked\n
*\n@SinceKotlin("1.2")\npublic fun <T> Sequence<T>.chunked(size: Int): Sequence<List<T>> {\n    return
windowed(size, size, partialWindows = true)\n}\n\n**\n * Splits this sequence into several lists each not exceeding
the given [size]\n * and applies the given [transform] function to an each.\n * \n * @return sequence of results of the
[transform] applied to an each list.\n * \n * Note that the list passed to the [transform] function is ephemeral and is
valid only inside that function.\n * You should not store it or allow it to escape in some way, unless you made a
snapshot of it.\n * The last list may have fewer elements than the given [size].\n * \n * @param size the number of
elements to take in each list, must be positive and can be greater than the number of elements in this sequence.\n * \n
* The operation is _intermediate_ and _stateful_.\n * \n * @sample samples.text.Strings.chunkedTransform\n
*\n@SinceKotlin("1.2")\npublic fun <T, R> Sequence<T>.chunked(size: Int, transform: (List<T>) -> R):
Sequence<R> {\n    return windowed(size, size, partialWindows = true, transform = transform)\n}\n\n**\n *
Returns a sequence containing all elements of the original sequence without the first occurrence of the given
[element].\n * \n * The operation is _intermediate_ and _stateless_.\n *\npublic operator fun <T>
Sequence<T>.minus(element: T): Sequence<T> {\n    return object: Sequence<T> {\n        override fun iterator():
Iterator<T> {\n            var removed = false\n            return this@minus.filter { if (!removed && it == element) {\n
                removed = true; false } else true }.iterator()\n        }\n    }\n}\n\n**\n * Returns a sequence containing all elements
of original sequence except the elements contained in the given [elements] array.\n * \n * Note that the source
sequence and the array being subtracted are iterated only when an `iterator` is requested from\n * the resulting
sequence. Changing any of them between successive calls to `iterator` may affect the result.\n * \n * Before Kotlin
1.6, the [elements] array may have been converted to a [HashSet] to speed up the operation, thus the elements were
required to have\n * a correct and stable implementation of `hashCode()` that didn't change between successive
invocations.\n * On JVM, you can enable this behavior back with the system property
`kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`.\n * \n * The operation is _intermediate_ and
_stateful_.\n *\npublic operator fun <T> Sequence<T>.minus(elements: Array<out T>): Sequence<T> {\n    if
(elements.isEmpty()) return this\n    return object: Sequence<T> {\n        override fun iterator(): Iterator<T> {\n
            val other = elements.convertToSetForSetOperation()\n            return this@minus.filterNot { it in other }.iterator()\n
        }\n    }\n}\n\n**\n * Returns a sequence containing all elements of original sequence except the elements
contained in the given [elements] collection.\n * \n * Note that the source sequence and the collection being
subtracted are iterated only when an `iterator` is requested from\n * the resulting sequence. Changing any of them
between successive calls to `iterator` may affect the result.\n * \n * Before Kotlin 1.6, the [elements] collection may
have been converted to a [HashSet] to speed up the operation, thus the elements were required to have\n * a correct
and stable implementation of `hashCode()` that didn't change between successive invocations.\n * On JVM, you can
enable this behavior back with the system property `kotlin.collections.convert_arg_to_set_in_removeAll` set to
`true`.\n * \n * The operation is _intermediate_ and _stateful_.\n *\npublic operator fun <T>
Sequence<T>.minus(elements: Iterable<T>): Sequence<T> {\n    return object: Sequence<T> {\n        override fun
iterator(): Iterator<T> {\n            val other = elements.convertToSetForSetOperation()\n            if (other.isEmpty())\n                return this@minus.iterator()\n            else\n                return this@minus.filterNot { it in other }.iterator()\n        }\n    }\n}\n

```



```

*^@SinceKotlin("1.2")\npublic fun <T> Sequence<T>.windowed(size: Int, step: Int = 1, partialWindows:
Boolean = false): Sequence<List<T>> {\n    return windowedSequence(size, step, partialWindows, reuseBuffer =
false)\n}\n\n/**\n * Returns a sequence of results of applying the given [transform] function to\n * an each list
representing a view over the window of the given [size]\n * sliding along this sequence with the given [step].\n * \n
* Note that the list passed to the [transform] function is ephemeral and is valid only inside that function.\n * You
should not store it or allow it to escape in some way, unless you made a snapshot of it.\n * Several last lists may
have fewer elements than the given [size].\n * \n * Both [size] and [step] must be positive and can be greater than
the number of elements in this sequence.\n * @param size the number of elements to take in each window\n *
@param step the number of elements to move the window forward by on an each step, by default 1\n * @param
partialWindows controls whether or not to keep partial windows in the end if any,\n * by default `false` which
means partial windows won't be preserved\n * \n * @sample
samples.collections.Sequences.Transformations.averageWindows\n *^@SinceKotlin("1.2")\npublic fun <T, R>
Sequence<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (List<T>) -> R):
Sequence<R> {\n    return windowedSequence(size, step, partialWindows, reuseBuffer =
true).map(transform)\n}\n\n/**\n * Returns a sequence of values built from the elements of `this` sequence and the
[other] sequence with the same index.\n * The resulting sequence ends as soon as the shortest input sequence ends.\n
*\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample
samples.collections.Sequences.Transformations.zip\n *^@SinceKotlin("1.2")\npublic infix fun <T, R> Sequence<T>.zip(other:
Sequence<R>): Sequence<Pair<T, R>> {\n    return MergingSequence(this, other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n *
Returns a sequence of values built from the elements of `this` sequence and the [other] sequence with the same
index\n * using the provided [transform] function applied to each pair of elements.\n * The resulting sequence ends
as soon as the shortest input sequence ends.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n *
@param sample samples.collections.Sequences.Transformations.zipWithTransform\n *^@SinceKotlin("1.2")\npublic fun <T, R, V>
Sequence<T>.zip(other: Sequence<R>, transform: (a: T, b: R) -> V): Sequence<V> {\n    return
MergingSequence(this, other, transform)\n}\n\n/**\n * Returns a sequence of pairs of each two adjacent elements in
this sequence.\n * \n * The returned sequence is empty if this sequence contains less than two elements.\n * \n * The
operation is _intermediate_ and _stateless_.\n * \n * @sample
samples.collections.Collections.Transformations.zipWithNext\n *^@SinceKotlin("1.2")\npublic fun <T>
Sequence<T>.zipWithNext(): Sequence<Pair<T, T>> {\n    return zipWithNext { a, b -> a to b }\n}\n\n/**\n *
Returns a sequence containing the results of applying the given [transform] function\n * to an each pair of two
adjacent elements in this sequence.\n * \n * The returned sequence is empty if this sequence contains less than two
elements.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample
samples.collections.Collections.Transformations.zipWithNextToFindDeltas\n *^@SinceKotlin("1.2")\npublic
fun <T, R> Sequence<T>.zipWithNext(transform: (a: T, b: T) -> R): Sequence<R> {\n    return sequence result@
{\n        val iterator = iterator()\n        if (!iterator.hasNext()) return@result\n        var current = iterator.next()\n
while (iterator.hasNext()) {\n            val next = iterator.next()\n            yield(transform(current, next))\n
current = next\n        }\n    }\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and
using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-
negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the
[truncated] string (which defaults to "...").\n *\n * The operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Transformations.joinTo\n *^@SinceKotlin("1.2")\npublic fun <T, A : Appendable>
Sequence<T>.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "\\", postfix:
CharSequence = "\\", limit: Int = -1, truncated: CharSequence = "...\", transform: ((T) -> CharSequence)? = null): A
{\n    buffer.append(prefix)\n    var count = 0\n    for (element in this) {\n        if (++count > 1)
buffer.append(separator)\n        if (limit < 0 || count <= limit) {\n            buffer.appendElement(element, transform)\n
        } else break\n    }\n    if (limit >= 0 && count > limit) buffer.append(truncated)\n    buffer.append(postfix)\n
return buffer\n}\n\n/**\n * Creates a string from all the elements separated using [separator] and using the given
[prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of

```


contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n

```
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("SetsKt")\n\npackage
```

```
kotlin.collections\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.*\nimport
```

```
kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns a set containing all elements of the original set except the given [element].\n * \n * The returned set preserves the element iteration order of the original set.\n
```

```
*/\n\npublic operator fun <T> Set<T>.minus(element: T): Set<T> {\n    val result =
```

```
LinkedHashSet<T>(mapCapacity(size))\n    var removed = false\n    return this.filterTo(result) { if (!removed && it == element) { removed = true; false } else true }\n}\n\n/**\n * Returns a set containing all elements of the original set except the elements contained in the given [elements] array.\n * \n * The returned set preserves the element iteration order of the original set.\n * \n * Before Kotlin 1.6, the [elements] array may have been converted to a [HashSet] to speed up the operation, thus the elements were required to have\n * a correct and stable implementation of `hashCode()` that didn't change between successive invocations.\n * On JVM, you can enable this behavior back with the system property `kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`.\n */\n\npublic operator fun <T> Set<T>.minus(elements: Array<out T>): Set<T> {\n    val result = LinkedHashSet<T>(this)\n    result.removeAll(elements)\n    return result\n}\n\n/**\n * Returns a set containing all elements of the original set except the elements contained in the given [elements] collection.\n * \n * The returned set preserves the element iteration order of the original set.\n * \n * Before Kotlin 1.6, the [elements] collection may have been converted to a [HashSet] to speed up the operation, thus the elements were required to have\n * a correct and stable implementation of `hashCode()` that didn't change between successive invocations.\n * On JVM, you can enable this behavior back with the system property `kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`.\n */\n\npublic operator fun <T> Set<T>.minus(elements: Iterable<T>): Set<T> {\n    val other = elements.convertToSetForSetOperationWith(this)\n    if (other.isEmpty())\n        return this.toSet()\n    if (other is Set)\n        return this.filterNotTo(LinkedHashSet<T>()) { it in other }\n    val result = LinkedHashSet<T>(this)\n    result.removeAll(other)\n    return result\n}\n\n/**\n * Returns a set containing all elements of the original set except the elements contained in the given [elements] sequence.\n * \n * The returned set preserves the element iteration order of the original set.\n * \n * Before Kotlin 1.6, the [elements] sequence may have been converted to a [HashSet] to speed up the operation, thus the elements were required to have\n * a correct and stable implementation of `hashCode()` that didn't change between successive invocations.\n * On JVM, you can enable this behavior back with the system property `kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`.\n */\n\npublic operator fun <T> Set<T>.minus(elements: Sequence<T>): Set<T> {\n    val result = LinkedHashSet<T>(this)\n    result.removeAll(elements)\n    return result\n}\n\n/**\n * Returns a set containing all elements of the original set and the given [elements] array,\n * which aren't already in this set.\n * \n * The returned set preserves the element iteration order of the original set.\n
```

```
*/\n\n@kotlin.internal.InlineOnly\npublic inline fun <T> Set<T>.minusElement(element: T): Set<T> {\n    return minus(element)\n}\n\n/**\n * Returns a set containing all elements of the original set and then the given [element] if it isn't already in this set.\n * \n * The returned set preserves the element iteration order of the original set.\n
```

```
*/\n\npublic operator fun <T> Set<T>.plus(element: T): Set<T> {\n    val result =
```

```
LinkedHashSet<T>(mapCapacity(size + 1))\n    result.addAll(this)\n    result.add(element)\n    return result\n}\n\n/**\n * Returns a set containing all elements of the original set and the given [elements] array,\n * which aren't already in this set.\n * \n * The returned set preserves the element iteration order of the original set.\n
```

```
*/\n\npublic operator fun <T> Set<T>.plus(elements: Array<out T>): Set<T> {\n    val result =
```

```
LinkedHashSet<T>(mapCapacity(this.size + elements.size))\n    result.addAll(this)\n    result.addAll(elements)\n    return result\n}\n\n/**\n * Returns a set containing all elements of the original set and the given [elements] collection,\n * which aren't already in this set.\n * \n * The returned set preserves the element iteration order of the original set.\n */\n\npublic operator fun <T> Set<T>.plus(elements: Iterable<T>): Set<T> {\n    val result =
```

```
LinkedHashSet<T>(mapCapacity(elements.collectionSizeOrNull()?.let { this.size + it } ?: this.size * 2))\n    result.addAll(this)\n    result.addAll(elements)\n    return result\n}\n\n/**\n * Returns a set containing all elements of
```

```
the original set and the given [elements] array,\n * which aren't already in this set.\n * \n * The returned set preserves the element iteration order of the original set.\n
```

```
*/\n\n@kotlin.internal.InlineOnly\npublic inline fun <T> Set<T>.minusElement(element: T): Set<T> {\n    return minus(element)\n}\n\n/**\n * Returns a set containing all elements of the original set and then the given [element] if it isn't already in this set.\n * \n * The returned set preserves the element iteration order of the original set.\n
```

```
*/\n\npublic operator fun <T> Set<T>.plus(element: T): Set<T> {\n    val result =
```

```
LinkedHashSet<T>(mapCapacity(size + 1))\n    result.addAll(this)\n    result.add(element)\n    return result\n}\n\n/**\n * Returns a set containing all elements of the original set and the given [elements] array,\n * which aren't already in this set.\n * \n * The returned set preserves the element iteration order of the original set.\n
```

```
*/\n\npublic operator fun <T> Set<T>.plus(elements: Array<out T>): Set<T> {\n    val result =
```

```
LinkedHashSet<T>(mapCapacity(this.size + elements.size))\n    result.addAll(this)\n    result.addAll(elements)\n    return result\n}\n\n/**\n * Returns a set containing all elements of the original set and the given [elements] collection,\n * which aren't already in this set.\n * \n * The returned set preserves the element iteration order of the original set.\n */\n\npublic operator fun <T> Set<T>.plus(elements: Iterable<T>): Set<T> {\n    val result =
```

```
LinkedHashSet<T>(mapCapacity(elements.collectionSizeOrNull()?.let { this.size + it } ?: this.size * 2))\n    result.addAll(this)\n    result.addAll(elements)\n    return result\n}\n\n/**\n * Returns a set containing all elements of
```

```
the original set and the given [elements] array,\n * which aren't already in this set.\n * \n * The returned set preserves the element iteration order of the original set.\n
```

```
*/\n\n@kotlin.internal.InlineOnly\npublic inline fun <T> Set<T>.minusElement(element: T): Set<T> {\n    return minus(element)\n}\n\n/**\n * Returns a set containing all elements of the original set and then the given [element] if it isn't already in this set.\n * \n * The returned set preserves the element iteration order of the original set.\n
```

```
*/\n\npublic operator fun <T> Set<T>.plus(element: T): Set<T> {\n    val result =
```

```
LinkedHashSet<T>(mapCapacity(size + 1))\n    result.addAll(this)\n    result.add(element)\n    return result\n}\n\n/**\n * Returns a set containing all elements of the original set and the given [elements] array,\n * which aren't already in this set.\n * \n * The returned set preserves the element iteration order of the original set.\n
```

```
*/\n\npublic operator fun <T> Set<T>.plus(elements: Array<out T>): Set<T> {\n    val result =
```

```
LinkedHashSet<T>(mapCapacity(this.size + elements.size))\n    result.addAll(this)\n    result.addAll(elements)\n    return result\n}\n\n/**\n * Returns a set containing all elements of the original set and the given [elements] collection,\n * which aren't already in this set.\n * \n * The returned set preserves the element iteration order of the original set.\n */\n\npublic operator fun <T> Set<T>.plus(elements: Iterable<T>): Set<T> {\n    val result =
```

```
LinkedHashSet<T>(mapCapacity(elements.collectionSizeOrNull()?.let { this.size + it } ?: this.size * 2))\n    result.addAll(this)\n    result.addAll(elements)\n    return result\n}\n\n/**\n * Returns a set containing all elements of
```

the original set and the given [elements] sequence, which aren't already in this set. The returned set preserves the element iteration order of the original set.

```

public operator fun <T> Set<T>.plus(elements:
Sequence<T>): Set<T> {
    val result = LinkedHashMap<T>(mapCapacity(this.size * 2))
    result.addAll(this)
    result.addAll(elements)
    return result
}

```

Returns a set containing all elements of the original set and then the given [element] if it isn't already in this set. The returned set preserves the element iteration order of the original set.

```

@kotlin.internal.InlineOnly
public inline fun <T> Set<T>.plusElement(element: T):
Set<T> {
    return plus(element)
}

```

Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors. Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

```

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("StringsKt")
package
kotlin.text
// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
import kotlin.random.*

```

Returns a character at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this char sequence.

```

@sample
samples.collections.Collections.Elements.elementAt

```

public expect fun CharSequence.elementAt(index: Int): Char Returns a character at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this char sequence.

```

@sample
samples.collections.Collections.Elements.elementAtOrElse

```

public inline fun CharSequence.elementAtOrElse(index: Int, defaultValue: (Int) -> Char): Char Returns a character at the given [index] or `null` if the [index] is out of bounds of this char sequence.

```

@sample
samples.collections.Collections.Elements.elementAtOrNull

```

public inline fun CharSequence.elementAtOrNull(index: Int): Char? Returns the first character matching the given [predicate], or `null` if no such character was found.

```

@sample
samples.collections.Collections.Elements.find

```

public inline fun CharSequence.find(predicate: (Char) -> Boolean): Char? Returns the last character matching the given [predicate], or `null` if no such character was found.

```

@sample
samples.collections.Collections.Elements.findLast

```

public inline fun CharSequence.findLast(predicate: (Char) -> Boolean): Char? Returns the first character matching the given [predicate], or `null` if no such character was found.

```

@throws NoSuchElementException if the char sequence is empty.
public fun CharSequence.first(): Char {
    if (isEmpty())
        throw NoSuchElementException("Char sequence is empty.")
    return this[0]
}

```

Returns the first character matching the given [predicate].

```

@throws [NoSuchElementException] if no such character is found.
public inline fun CharSequence.first(predicate:
(Char) -> Boolean): Char {
    for (element in this)
        if (predicate(element))
            return element
    throw NoSuchElementException("Char sequence contains no character matching the predicate.")
}

```

Returns the first non-null value produced by [transform] function being applied to characters of this char sequence in iteration order, or throws [NoSuchElementException] if no non-null value was produced.

```

@sample
samples.collections.Collections.Transformations.firstNotNullOf

```

Since Kotlin("1.5")

```

@kotlin.internal.InlineOnly
public inline fun <R : Any>
CharSequence.firstNotNullOf(transform: (Char) -> R?): R {
    return firstNotNullOfOrNull(transform)?: throw
NoSuchElementException("No element of the char sequence was transformed to a non-null value.")
}

```

Returns the first non-null value produced by [transform] function being applied to characters of this char sequence in iteration order, or `null` if no non-null value was produced.

```

@sample
samples.collections.Collections.Transformations.firstNotNullOf

```

Since Kotlin("1.5")

```

@kotlin.internal.InlineOnly
public inline fun <R : Any>
CharSequence.firstNotNullOfOrNull(transform: (Char) -> R?): R? {
    for (element in this) {
        val result =
transform(element)
        if (result != null)
            return result
    }
    return null
}

```

Returns the first character, or `null` if the char sequence is empty.

```

public fun CharSequence.firstOrNull():
Char? {
    return if (isEmpty())
        null
    else
        this[0]
}

```

Returns the first character matching the given

```

[predicate], or `null` if character was not found.\n */\npublic inline fun CharSequence.firstOrNull(predicate: (Char) -
> Boolean): Char? {\n    for (element in this) if (predicate(element)) return element\n    return null\n}\n\n/**\n *
Returns a character at the given [index] or the result of calling the [defaultValue] function if the [index] is out of
bounds of this char sequence.\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.getOrElse(index:
Int, defaultValue: (Int) -> Char): Char {\n    return if (index >= 0 && index <= lastIndex) get(index) else
defaultValue(index)\n}\n\n/**\n * Returns a character at the given [index] or `null` if the [index] is out of bounds of
this char sequence.\n */\n * @sample samples.collections.Collections.Elements.getOrElse\n */\npublic fun
CharSequence.getOrElse(index: Int): Char? {\n    return if (index >= 0 && index <= lastIndex) get(index) else
null\n}\n\n/**\n * Returns index of the first character matching the given [predicate], or -1 if the char sequence does
not contain such character.\n */\npublic inline fun CharSequence.indexOfFirst(predicate: (Char) -> Boolean): Int {\n
    for (index in indices) {\n        if (predicate(this[index])) {\n            return index\n        }\n    }\n    return -
1\n}\n\n/**\n * Returns index of the last character matching the given [predicate], or -1 if the char sequence does
not contain such character.\n */\npublic inline fun CharSequence.indexOfLast(predicate: (Char) -> Boolean): Int {\n
    for (index in indices.reversed()) {\n        if (predicate(this[index])) {\n            return index\n        }\n    }\n    return -
1\n}\n\n/**\n * Returns the last character.\n */\n * @throws NoSuchElementException if the char sequence is
empty.\n */\n * @sample samples.text.Strings.last\n */\npublic fun CharSequence.last(): Char {\n    if (isEmpty())\n        throw NoSuchElementException("Char sequence is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the
last character matching the given [predicate].\n */\n * @throws NoSuchElementException if no such character is
found.\n */\n * @sample samples.text.Strings.last\n */\npublic inline fun CharSequence.last(predicate: (Char) ->
Boolean): Char {\n    for (index in this.indices.reversed()) {\n        val element = this[index]\n        if
(predicate(element)) return element\n    }\n    throw NoSuchElementException("Char sequence contains no
character matching the predicate.")\n}\n\n/**\n * Returns the last character, or `null` if the char sequence is
empty.\n */\n * @sample samples.text.Strings.last\n */\npublic fun CharSequence.lastOrNull(): Char? {\n    return if
(isEmpty()) null else this[length - 1]\n}\n\n/**\n * Returns the last character matching the given [predicate], or `null`
if no such character was found.\n */\n * @sample samples.text.Strings.last\n */\npublic inline fun
CharSequence.lastOrNull(predicate: (Char) -> Boolean): Char? {\n    for (index in this.indices.reversed()) {\n
        val element = this[index]\n        if (predicate(element)) return element\n    }\n    return null\n}\n\n/**\n * Returns a
random character from this char sequence.\n */\n * @throws NoSuchElementException if this char sequence is
empty.\n */\n * @SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.random(): Char
{\n    return random(Random)\n}\n\n/**\n * Returns a random character from this char sequence using the specified
source of randomness.\n */\n * @throws NoSuchElementException if this char sequence is empty.\n */\n * @SinceKotlin("1.3")\npublic fun CharSequence.random(random: Random): Char {\n    if (isEmpty())\n        throw NoSuchElementException("Char sequence is empty.")\n    return get(random.nextInt(length))\n}\n\n/**\n * Returns a random character from this char sequence, or `null` if this char sequence is empty.\n */\n * @SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun CharSequence.randomOrNull(): Char? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random character from this char sequence using the specified source of randomness, or `null` if this char sequence is
empty.\n */\n * @SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
CharSequence.randomOrNull(random: Random): Char? {\n    if (isEmpty())\n        return null\n    return
get(random.nextInt(length))\n}\n\n/**\n * Returns the single character, or throws an exception if the char sequence
is empty or has more than one character.\n */\npublic fun CharSequence.single(): Char {\n    return when (length)
{\n        0 -> throw NoSuchElementException("Char sequence is empty.")\n        1 -> this[0]\n        else -> throw
IllegalArgumentException("Char sequence has more than one element.")\n    }\n}\n\n/**\n * Returns the single
character matching the given [predicate], or throws exception if there is no or more than one matching character.\n */\n * @public inline fun CharSequence.single(predicate: (Char) -> Boolean): Char {\n    var single: Char? = null\n    var
found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) throw
IllegalArgumentException("Char sequence contains more than one matching element.")\n            single =
element\n            found = true\n        }\n    }\n    if (!found) throw NoSuchElementException("Char sequence

```

```

contains no character matching the predicate.)\n  @Suppress("UNCHECKED_CAST")\n  return single as
Char\n}\n\n/**\n * Returns single character, or `null` if the char sequence is empty or has more than one character.\n
*/\npublic fun CharSequence.singleOrNull(): Char? {\n  return if (length == 1) this[0] else null\n}\n\n/**\n *
Returns the single character matching the given [predicate], or `null` if character was not found or more than one
character was found.\n */\npublic inline fun CharSequence.singleOrNull(predicate: (Char) -> Boolean): Char? {\n
var single: Char? = null\n  var found = false\n  for (element in this) {\n    if (predicate(element)) {\n      if
(found) return null\n      single = element\n      found = true\n    }\n  }\n  if (!found) return null\n  return
single\n}\n\n/**\n * Returns a subsequence of this char sequence with the first [n] characters removed.\n */\n
@throws IllegalArgumentException if [n] is negative.\n */\n * @sample samples.text.Strings.drop\n */\npublic fun
CharSequence.drop(n: Int): CharSequence {\n  require(n >= 0) { "Requested character count $n is less than zero."
}\n  return subSequence(n.coerceAtMost(length), length)\n}\n\n/**\n * Returns a string with the first [n] characters
removed.\n */\n * @throws IllegalArgumentException if [n] is negative.\n */\n * @sample
samples.text.Strings.drop\n */\npublic fun String.drop(n: Int): String {\n  require(n >= 0) { "Requested character
count $n is less than zero." }\n  return substring(n.coerceAtMost(length))\n}\n\n/**\n * Returns a subsequence of
this char sequence with the last [n] characters removed.\n */\n * @throws IllegalArgumentException if [n] is
negative.\n */\n * @sample samples.text.Strings.drop\n */\npublic fun CharSequence.dropLast(n: Int):
CharSequence {\n  require(n >= 0) { "Requested character count $n is less than zero." }\n  return take((length -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a string with the last [n] characters removed.\n */\n * @throws
IllegalArgumentException if [n] is negative.\n */\n * @sample samples.text.Strings.drop\n */\npublic fun
String.dropLast(n: Int): String {\n  require(n >= 0) { "Requested character count $n is less than zero." }\n  return
take((length - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a subsequence of this char sequence containing all
characters except last characters that satisfy the given [predicate].\n */\n * @sample samples.text.Strings.drop\n
*/\npublic inline fun CharSequence.dropLastWhile(predicate: (Char) -> Boolean): CharSequence {\n  for (index in
lastIndex downTo 0)\n    if (!predicate(this[index]))\n      return subSequence(0, index + 1)\n  return
""\n}\n\n/**\n * Returns a string containing all characters except last characters that satisfy the given [predicate].\n
*/\n * @sample samples.text.Strings.drop\n */\npublic inline fun String.dropLastWhile(predicate: (Char) ->
Boolean): String {\n  for (index in lastIndex downTo 0)\n    if (!predicate(this[index]))\n      return
substring(0, index + 1)\n  return ""\n}\n\n/**\n * Returns a subsequence of this char sequence containing all
characters except first characters that satisfy the given [predicate].\n */\n * @sample samples.text.Strings.drop\n
*/\npublic inline fun CharSequence.dropWhile(predicate: (Char) -> Boolean): CharSequence {\n  for (index in
this.indices)\n    if (!predicate(this[index]))\n      return subSequence(index, length)\n  return ""\n}\n\n/**\n
* Returns a string containing all characters except first characters that satisfy the given [predicate].\n */\n *
@sample samples.text.Strings.drop\n */\npublic inline fun String.dropWhile(predicate: (Char) -> Boolean): String
{\n  for (index in this.indices)\n    if (!predicate(this[index]))\n      return substring(index)\n  return
""\n}\n\n/**\n * Returns a char sequence containing only those characters from the original char sequence that
match the given [predicate].\n */\n * @sample samples.text.Strings.filter\n */\npublic inline fun
CharSequence.filter(predicate: (Char) -> Boolean): CharSequence {\n  return filterTo(StringBuilder(),
predicate)\n}\n\n/**\n * Returns a string containing only those characters from the original string that match the
given [predicate].\n */\n * @sample samples.text.Strings.filter\n */\npublic inline fun String.filter(predicate: (Char) -
> Boolean): String {\n  return filterTo(StringBuilder(), predicate).toString()\n}\n\n/**\n * Returns a char sequence
containing only those characters from the original char sequence that match the given [predicate].\n */\n * @param
[predicate] function that takes the index of a character and the character itself\n * and returns the result of predicate
evaluation on the character.\n */\n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic
inline fun CharSequence.filterIndexed(predicate: (index: Int, Char) -> Boolean): CharSequence {\n  return
filterIndexedTo(StringBuilder(), predicate)\n}\n\n/**\n * Returns a string containing only those characters from the
original string that match the given [predicate].\n */\n * @param [predicate] function that takes the index of a character
and the character itself\n * and returns the result of predicate evaluation on the character.\n */\n * @sample
samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun String.filterIndexed(predicate: (index:

```



```

Int, Char) -> Boolean): String {\n    return filterIndexedTo(StringBuilder(), predicate).toString()\n}\n\n/**\n *
Appends all characters matching the given [predicate] to the given [destination].\n * @param [predicate] function
that takes the index of a character and the character itself\n * and returns the result of predicate evaluation on the
character.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C :
Appendable> CharSequence.filterIndexedTo(destination: C, predicate: (index: Int, Char) -> Boolean): C {\n
forEachIndexed { index, element ->\n    if (predicate(index, element)) destination.append(element)\n    }\n
return destination\n}\n\n/**\n * Returns a char sequence containing only those characters from the original char
sequence that do not match the given [predicate].\n * \n * @sample samples.text.Strings.filterNot\n */\npublic inline
fun CharSequence.filterNot(predicate: (Char) -> Boolean): CharSequence {\n    return filterNotTo(StringBuilder(),
predicate)\n}\n\n/**\n * Returns a string containing only those characters from the original string that do not match
the given [predicate].\n * \n * @sample samples.text.Strings.filterNot\n */\npublic inline fun
String.filterNot(predicate: (Char) -> Boolean): String {\n    return filterNotTo(StringBuilder(),
predicate).toString()\n}\n\n/**\n * Appends all characters not matching the given [predicate] to the given
[destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C :
Appendable> CharSequence.filterNotTo(destination: C, predicate: (Char) -> Boolean): C {\n    for (element in this)
if (!predicate(element)) destination.append(element)\n    return destination\n}\n\n/**\n * Appends all characters
matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : Appendable>
CharSequence.filterTo(destination: C, predicate: (Char) -> Boolean): C {\n    for (index in 0 until length) {\n        val
element = get(index)\n        if (predicate(element)) destination.append(element)\n    }\n    return
destination\n}\n\n/**\n * Returns a char sequence containing characters of the original char sequence at the
specified range of [indices].\n */\npublic fun CharSequence.slice(indices: IntRange): CharSequence {\n    if
(indices.isEmpty()) return ""\n    return subSequence(indices)\n}\n\n/**\n * Returns a string containing characters
of the original string at the specified range of [indices].\n */\npublic fun String.slice(indices: IntRange): String {\n
if (indices.isEmpty()) return ""\n    return substring(indices)\n}\n\n/**\n * Returns a char sequence containing
characters of the original char sequence at specified [indices].\n */\npublic fun CharSequence.slice(indices:
Iterable<Int>): CharSequence {\n    val size = indices.collectionSizeOrDefault(10)\n    if (size == 0) return ""\n
val result = StringBuilder(size)\n    for (i in indices) {\n        result.append(get(i))\n    }\n    return result\n}\n\n/**\n * Returns a string containing characters of the original string at specified [indices].\n
*/\n\n@kotlin.internal.InlineOnly\npublic inline fun String.slice(indices: Iterable<Int>): String {\n    return (this as
CharSequence).slice(indices).toString()\n}\n\n/**\n * Returns a subsequence of this char sequence containing the
first [n] characters from this char sequence, or the entire char sequence if this char sequence is shorter.\n * \n *
@throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.text.Strings.take\n */\npublic fun
CharSequence.take(n: Int): CharSequence {\n    require(n >= 0) { "Requested character count $n is less than zero."
}\n    return subSequence(0, n.coerceAtMost(length))\n}\n\n/**\n * Returns a string containing the first [n]
characters from this string, or the entire string if this string is shorter.\n * \n * @throws IllegalArgumentException if
[n] is negative.\n * \n * @sample samples.text.Strings.take\n */\npublic fun String.take(n: Int): String {\n    require(n
>= 0) { "Requested character count $n is less than zero." }\n    return substring(0,
n.coerceAtMost(length))\n}\n\n/**\n * Returns a subsequence of this char sequence containing the last [n]
characters from this char sequence, or the entire char sequence if this char sequence is shorter.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample samples.text.Strings.take\n */\npublic fun
CharSequence.takeLast(n: Int): CharSequence {\n    require(n >= 0) { "Requested character count $n is less than
zero." }\n    val length = length\n    return subSequence(length - n.coerceAtMost(length), length)\n}\n\n/**\n *
Returns a string containing the last [n] characters from this string, or the entire string if this string is shorter.\n *
\n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.text.Strings.take\n */\npublic fun
String.takeLast(n: Int): String {\n    require(n >= 0) { "Requested character count $n is less than zero." }\n    val
length = length\n    return substring(length - n.coerceAtMost(length))\n}\n\n/**\n * Returns a subsequence of this
char sequence containing last characters that satisfy the given [predicate].\n * \n * @sample

```

```

samples.text.Strings.take\n */\npublic inline fun CharSequence.takeLastWhile(predicate: (Char) -> Boolean):
CharSequence {\n  for (index in lastIndex downTo 0) {\n    if (!predicate(this[index])) {\n      return
subSequence(index + 1, length)\n    }\n  }\n  return subSequence(0, length)\n}\n\n/**\n * Returns a string
containing last characters that satisfy the given [predicate].\n * \n * @sample samples.text.Strings.take\n */\npublic
inline fun String.takeLastWhile(predicate: (Char) -> Boolean): String {\n  for (index in lastIndex downTo 0) {\n
if (!predicate(this[index])) {\n    return substring(index + 1)\n  }\n}\n  return this\n}\n\n/**\n * Returns
a subsequence of this char sequence containing the first characters that satisfy the given [predicate].\n * \n *
@sample samples.text.Strings.take\n */\npublic inline fun CharSequence.takeWhile(predicate: (Char) -> Boolean):
CharSequence {\n  for (index in 0 until length)\n    if (!predicate(get(index))) {\n      return subSequence(0,
index)\n    }\n  }\n  return subSequence(0, length)\n}\n\n/**\n * Returns a string containing the first characters that
satisfy the given [predicate].\n * \n * @sample samples.text.Strings.take\n */\npublic inline fun
String.takeWhile(predicate: (Char) -> Boolean): String {\n  for (index in 0 until length)\n    if
(!predicate(get(index))) {\n      return substring(0, index)\n    }\n  }\n  return this\n}\n\n/**\n * Returns a char
sequence with characters in reversed order.\n */\npublic fun CharSequence.reversed(): CharSequence {\n  return
StringBuilder(this).reverse()\n}\n\n/**\n * Returns a string with characters in reversed order.\n */\n@kotlin.internal.InlineOnly\npublic inline fun String.reversed(): String {\n  return (this as
CharSequence).reversed().toString()\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by
[transform] function\n * applied to characters of the given char sequence.\n * \n * If any of two pairs would have the
same key the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the
original char sequence.\n * \n * @sample samples.text.Strings.associate\n */\npublic inline fun <K, V>
CharSequence.associate(transform: (Char) -> Pair<K, V>): Map<K, V> {\n  val capacity =
mapCapacity(length).coerceAtLeast(16)\n  return associateTo(LinkedHashMap<K, V>(capacity),
transform)\n}\n\n/**\n * Returns a [Map] containing the characters from the given char sequence indexed by the
key\n * returned from [keySelector] function applied to each character.\n * \n * If any two characters would have the
same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry
iteration order of the original char sequence.\n * \n * @sample samples.text.Strings.associateBy\n */\npublic inline
fun <K> CharSequence.associateBy(keySelector: (Char) -> K): Map<K, Char> {\n  val capacity =
mapCapacity(length).coerceAtLeast(16)\n  return associateByTo(LinkedHashMap<K, Char>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform] and indexed by
[keySelector] functions applied to characters of the given char sequence.\n * \n * If any two characters would have
the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the
entry iteration order of the original char sequence.\n * \n * @sample
samples.text.Strings.associateByWithValueTransform\n */\npublic inline fun <K, V>
CharSequence.associateBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, V> {\n  val capacity
= mapCapacity(length).coerceAtLeast(16)\n  return associateByTo(LinkedHashMap<K, V>(capacity),
keySelector, valueTransform)\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value
pairs,\n * where key is provided by the [keySelector] function applied to each character of the given char sequence\n
* and value is the character itself.\n * \n * If any two characters would have the same key returned by [keySelector]
the last one gets added to the map.\n * \n * @sample samples.text.Strings.associateByTo\n */\npublic inline fun <K,
M : MutableMap<in K, in Char>> CharSequence.associateByTo(destination: M, keySelector: (Char) -> K): M {\n
for (element in this) {\n  destination.put(keySelector(element), element)\n }\n  return destination\n}\n\n/**\n *
Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the
[keySelector] function and\n * and value is provided by the [valueTransform] function applied to characters of the
given char sequence.\n * \n * If any two characters would have the same key returned by [keySelector] the last one
gets added to the map.\n * \n * @sample samples.text.Strings.associateByToWithValueTransform\n */\npublic
inline fun <K, V, M : MutableMap<in K, in V>> CharSequence.associateByTo(destination: M, keySelector: (Char)
-> K, valueTransform: (Char) -> V): M {\n  for (element in this) {\n    destination.put(keySelector(element),
valueTransform(element))\n  }\n  return destination\n}\n\n/**\n * Populates and returns the [destination] mutable

```

map with key-value pairs provided by [transform] function applied to each character of the given char sequence. If any of two pairs would have the same key the last one gets added to the map. @sample

```

samples.text.Strings.associateTo
public inline fun <K, V, M : MutableMap<in K, in V>>
CharSequence.associateTo(destination: M, transform: (Char) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}
Returns a [Map] where keys are characters from the given char sequence and values are produced by the [valueSelector] function applied to each character. If any two characters are equal, the last one gets added to the map. The returned map preserves the entry iteration order of the original char sequence. @sample

```

samples.text.Strings.associateWith

```

@SinceKotlin("1.3")
public inline fun <V>
CharSequence.associateWith(valueSelector: (Char) -> V): Map<Char, V> {
    val result = LinkedHashMap<Char, V>(mapCapacity(length.coerceAtMost(128)).coerceAtLeast(16))
    return associateWithTo(result, valueSelector)
}
Populates and returns the [destination] mutable map with key-value pairs for each character of the given char sequence, where key is the character itself and value is provided by the [valueSelector] function applied to that key. If any two characters are equal, the last one overwrites the former value in the map. @sample

```

samples.text.Strings.associateWithTo

```

@SinceKotlin("1.3")
public inline fun <V, M : MutableMap<in Char, in V>>
CharSequence.associateWithTo(destination: M, valueSelector: (Char) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}
Appends all characters to the given [destination] collection.
public fun <C : MutableCollection<in Char>>
CharSequence.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}
Returns a new [HashSet] of all characters.
public fun CharSequence.toHashSet(): HashSet<Char> {
    return toCollection(HashSet<Char>(mapCapacity(length.coerceAtMost(128))))
}
Returns a [List] containing all characters.
public fun CharSequence.toList(): List<Char> {
    return when (length) {
        0 -> emptyList()
        1 -> listOf(this[0])
        else -> this.toMutableList()
    }
}
Returns a new [MutableList] filled with all characters of this char sequence.
public fun
CharSequence.toMutableList(): MutableList<Char> {
    return toCollection(ArrayList<Char>(length))
}
Returns a [Set] of all characters. The returned set preserves the element iteration order of the original char sequence.
public fun CharSequence.toSet(): Set<Char> {
    return when (length) {
        0 -> emptySet()
        1 -> setOf(this[0])
        else -> toCollection(LinkedHashSet<Char>(mapCapacity(length.coerceAtMost(128))))
    }
}
Returns a single list of all elements yielded from results of [transform] function being invoked on each character of original char sequence. @sample

```

samples.collections.Collections.Transformations.flatMap

```

public inline fun <R>
CharSequence.flatMap(transform: (Char) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}
Returns a single list of all elements yielded from results of [transform] function being invoked on each character and its index in the original char sequence. @sample

```

samples.collections.Collections.Transformations.flatMapIndexed

```

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterable")
@kotlin.internal.InlineOnly
public inline fun <R>
CharSequence.flatMapIndexed(transform: (index: Int, Char) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}
Appends all elements yielded from results of [transform] function being invoked on each character and its index in the original char sequence, to the given [destination].

```

```

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterableTo")
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>>
CharSequence.flatMapIndexedTo(destination: C, transform: (index: Int, Char) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}
Appends all elements yielded from results of [transform] function being invoked on each character of original char sequence, to the given

```

```

[destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> CharSequence.flatMapTo(destination: C,
transform: (Char) -> Iterable<R>): C {\n for (element in this) {\n val list = transform(element)\n
destination.addAll(list)\n } \n return destination\n}\n\n/**\n * Groups characters of the original char sequence by
the key returned by the given [keySelector] function\n * applied to each character and returns a map where each
group key is associated with a list of corresponding characters.\n * \n * The returned map preserves the entry
iteration order of the keys produced from the original char sequence.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K>
CharSequence.groupBy(keySelector: (Char) -> K): Map<K, List<Char>> {\n return
groupByTo(LinkedHashMap<K, MutableList<Char>>(), keySelector)\n}\n\n/**\n * Groups values returned by the
[valueTransform] function applied to each character of the original char sequence\n * by the key returned by the
given [keySelector] function applied to the character\n * and returns a map where each group key is associated with
a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced
from the original char sequence.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V>
CharSequence.groupBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, List<V>> {\n return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups
characters of the original char sequence by the key returned by the given [keySelector] function\n * applied to each
character and puts to the [destination] map each group key associated with a list of corresponding characters.\n * \n
* @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*/\npublic inline fun <K, M : MutableMap<in K, MutableList<Char>>> CharSequence.groupByTo(destination: M,
keySelector: (Char) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<Char>() }\n list.add(element)\n } \n return destination\n}\n\n/**\n *
Groups values returned by the [valueTransform] function applied to each character of the original char sequence\n *
by the key returned by the given [keySelector] function applied to the character\n * and puts to the [destination] map
each group key associated with a list of corresponding values.\n * \n * @return The [destination] map.\n * \n *
@sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V,
M : MutableMap<in K, MutableList<V>>> CharSequence.groupByTo(destination: M, keySelector: (Char) -> K,
valueTransform: (Char) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n } \n return
destination\n}\n\n/**\n * Creates a [Grouping] source from a char sequence to be used later with one of group-and-
fold operations\n * using the specified [keySelector] function to extract a key from each character.\n * \n * @sample
samples.collections.Grouping.groupingByEachCount\n */\n@SinceKotlin("1.1")\npublic inline fun <K>
CharSequence.groupingBy(crossinline keySelector: (Char) -> K): Grouping<Char, K> {\n return object :
Grouping<Char, K> {\n override fun sourceIterator(): Iterator<Char> = this@groupingBy.iterator()\n
override fun keyOf(element: Char): K = keySelector(element)\n } \n}\n\n/**\n * Returns a list containing the
results of applying the given [transform] function\n * to each character in the original char sequence.\n * \n *
@sample samples.text.Strings.map\n */\npublic inline fun <R> CharSequence.map(transform: (Char) -> R):
List<R> {\n return mapTo(ArrayList<R>(length), transform)\n}\n\n/**\n * Returns a list containing the results of
applying the given [transform] function\n * to each character and its index in the original char sequence.\n *
@param [transform] function that takes the index of a character and the character itself\n * and returns the result of
the transform applied to the character.\n */\npublic inline fun <R> CharSequence.mapIndexed(transform: (index:
Int, Char) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(length), transform)\n}\n\n/**\n * Returns a list
containing only the non-null results of applying the given [transform] function\n * to each character and its index in
the original char sequence.\n * @param [transform] function that takes the index of a character and the character
itself\n * and returns the result of the transform applied to the character.\n */\npublic inline fun <R : Any>
CharSequence.mapIndexedNotNull(transform: (index: Int, Char) -> R?): List<R> {\n return
mapIndexedNotNullTo(ArrayList<R>(), transform)\n}\n\n/**\n * Applies the given [transform] function to each
character and its index in the original char sequence\n * and appends only the non-null results to the given

```

```

[destination].\n * @param [transform] function that takes the index of a character and the character itself\n * and
returns the result of the transform applied to the character.\n */\npublic inline fun <R : Any, C :
MutableCollection<in R>> CharSequence.mapIndexedNotNullTo(destination: C, transform: (index: Int, Char) ->
R?): C {\n    forEachIndexed { index, element -> transform(index, element)?.let { destination.add(it) } }\n    return
destination\n}\n\n/**\n * Applies the given [transform] function to each character and its index in the original char
sequence\n * and appends the results to the given [destination].\n * @param [transform] function that takes the
index of a character and the character itself\n * and returns the result of the transform applied to the character.\n
*/\npublic inline fun <R, C : MutableCollection<in R>> CharSequence.mapIndexedTo(destination: C, transform:
(index: Int, Char) -> R): C {\n    var index = 0\n    for (item in this)\n        destination.add(transform(index++,
item))\n    return destination\n}\n\n/**\n * Returns a list containing only the non-null results of applying the given
[transform] function\n * to each character in the original char sequence.\n * \n * @sample
samples.collections.Collections.Transformations.mapNotNull\n */\npublic inline fun <R : Any>
CharSequence.mapNotNull(transform: (Char) -> R?): List<R> {\n    return mapNotNullTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Applies the given [transform] function to each character in the original char sequence\n *
and appends only the non-null results to the given [destination].\n */\npublic inline fun <R : Any, C :
MutableCollection<in R>> CharSequence.mapNotNullTo(destination: C, transform: (Char) -> R?): C {\n    forEach
{ element -> transform(element)?.let { destination.add(it) } }\n    return destination\n}\n\n/**\n * Applies the given
[transform] function to each character of the original char sequence\n * and appends the results to the given
[destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> CharSequence.mapTo(destination: C,
transform: (Char) -> R): C {\n    for (item in this)\n        destination.add(transform(item))\n    return
destination\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each character of the original char sequence\n * into an
[IndexValue] containing the index of that character and the character itself.\n */\npublic fun
CharSequence.withIndex(): Iterable<IndexedValue<Char>> {\n    return IndexingIterable { iterator() }\n}\n\n/**\n *
Returns `true` if all characters match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.all\n */\npublic inline fun CharSequence.all(predicate: (Char) ->
Boolean): Boolean {\n    for (element in this) if (!predicate(element)) return false\n    return true\n}\n\n/**\n *
Returns `true` if char sequence has at least one character.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n */\npublic fun CharSequence.any(): Boolean {\n    return
!isEmpty()\n}\n\n/**\n * Returns `true` if at least one character matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n */\npublic inline fun CharSequence.any(predicate:
(Char) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return true\n    return
false\n}\n\n/**\n * Returns the length of this char sequence.\n */\n@kotlin.internal.InlineOnly\npublic inline fun
CharSequence.count(): Int {\n    return length\n}\n\n/**\n * Returns the number of characters matching the given
[predicate].\n */\npublic inline fun CharSequence.count(predicate: (Char) -> Boolean): Int {\n    var count = 0\n
for (element in this) if (predicate(element)) ++count\n    return count\n}\n\n/**\n * Accumulates value starting with
[initial] value and applying [operation] from left to right\n * to current accumulator value and each character.\n *
\n * Returns the specified [initial] value if the char sequence is empty.\n * \n * @param [operation] function that takes
current accumulator value and a character, and calculates the next accumulator value.\n */\npublic inline fun <R>
CharSequence.fold(initial: R, operation: (acc: R, Char) -> R): R {\n    var accumulator = initial\n    for (element in
this) accumulator = operation(accumulator, element)\n    return accumulator\n}\n\n/**\n * Accumulates value
starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each
character with its index in the original char sequence.\n * \n * Returns the specified [initial] value if the char
sequence is empty.\n * \n * @param [operation] function that takes the index of a character, current accumulator
value\n * and the character itself, and calculates the next accumulator value.\n */\npublic inline fun <R>
CharSequence.foldIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): R {\n    var index = 0\n    var
accumulator = initial\n    for (element in this) accumulator = operation(index++, accumulator, element)\n    return
accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to
left\n * to each character and current accumulator value.\n * \n * Returns the specified [initial] value if the char

```

sequence is empty.

```

    @param [operation] function that takes a character and current accumulator value, and
    calculates the next accumulator value.
    public inline fun <R> CharSequence.foldRight(initial: R, operation:
    (Char, acc: R) -> R): R {
        var index = lastIndex
        var accumulator = initial
        while (index >= 0) {
            accumulator = operation(get(index--), accumulator)
        }
        return accumulator
    }

```

Accumulates value starting with [initial] value and applying [operation] from right to left to each character with its index in the original char sequence and current accumulator value.

```

    Returns the specified [initial] value if the char
    sequence is empty.
    @param [operation] function that takes the index of a character, the character itself
    and current accumulator value, and calculates the next accumulator value.
    public inline fun <R>
    CharSequence.foldRightIndexed(initial: R, operation: (index: Int, Char, acc: R) -> R): R {
        var index =
        lastIndex
        var accumulator = initial
        while (index >= 0) {
            accumulator = operation(index, get(index),
            accumulator)
            --index
        }
        return accumulator
    }

```

Performs the given [action] on each character.

```

    public inline fun CharSequence.forEach(action: (Char) -> Unit): Unit {
        for (element in this)
            action(element)
    }

```

Performs the given [action] on each character, providing sequential index with the character.

```

    @param [action] function that takes the index of a character and the character itself
    and performs
    the action on the character.
    public inline fun CharSequence.forEachIndexed(action: (index: Int, Char) -> Unit):
    Unit {
        var index = 0
        for (item in this) action(index++, item)
    }

```

Returns the largest character.

```

    @throws NoSuchElementException if the char sequence is empty.
    @SinceKotlin("1.7")
    @kotlin.jvm.JvmName("maxOrThrow")
    @Suppress("CONFLICTING_OVERLOADS")
    public fun CharSequence.max(): Char {
        if (isEmpty()) throw NoSuchElementException()
        var max =
        this[0]
        for (i in 1..lastIndex) {
            val e = this[i]
            if (max < e) max = e
        }
        return max
    }

```

Returns the first character yielding the largest value of the given function.

```

    @throws
    NoSuchElementException if the char sequence is empty.
    @sample
    samples.collections.Collections.Aggregates.maxBy
    @SinceKotlin("1.7")
    @kotlin.jvm.JvmName("maxByOrThrow")
    @Suppress("CONFLICTING_OVERLOADS")
    public inline fun <R : Comparable<R>> CharSequence.maxBy(selector: (Char) -> R): Char {
        if
        (isEmpty()) throw NoSuchElementException()
        var maxElem = this[0]
        val lastIndex = this.lastIndex
        if
        (lastIndex == 0) return maxElem
        var maxValue = selector(maxElem)
        for (i in 1..lastIndex) {
            val e =
            this[i]
            val v = selector(e)
            if (maxValue < v) {
                maxElem = e
                maxValue = v
            }
        }
        return maxElem
    }

```

Returns the first character yielding the largest value of the given function or `null` if there are no characters.

```

    @sample samples.collections.Collections.Aggregates.maxByOrNull
    @SinceKotlin("1.4")
    public inline fun <R : Comparable<R>> CharSequence.maxByOrNull(selector: (Char) ->
    R): Char? {
        if (isEmpty()) return null
        var maxElem = this[0]
        val lastIndex = this.lastIndex
        if
        (lastIndex == 0) return maxElem
        var maxValue = selector(maxElem)
        for (i in 1..lastIndex) {
            val e =
            this[i]
            val v = selector(e)
            if (maxValue < v) {
                maxElem = e
                maxValue = v
            }
        }
        return maxElem
    }

```

Returns the largest value among all values produced by [selector] function applied to each character in the char sequence.

```

    If any of values produced by [selector] function is `NaN`, the
    returned result is `NaN`.
    @throws NoSuchElementException if the char sequence is empty.
    @SinceKotlin("1.4")
    @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
    @OverloadResolution
    ByLambdaReturnType
    @kotlin.internal.InlineOnly
    public inline fun CharSequence.maxOf(selector: (Char) ->
    Double): Double {
        if (isEmpty()) throw NoSuchElementException()
        var maxValue = selector(this[0])
        for
        (i in 1..lastIndex) {
            val v = selector(this[i])
            maxValue = maxOf(maxValue, v)
        }
        return
        maxValue
    }

```

Returns the largest value among all values produced by [selector] function applied to each character in the char sequence.

```

    If any of values produced by [selector] function is `NaN`, the returned
    result is `NaN`.
    @throws NoSuchElementException if the char sequence is empty.
    @SinceKotlin("1.4")
    @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
    @OverloadResolution
    ByLambdaReturnType
    @kotlin.internal.InlineOnly
    public inline fun CharSequence.maxOf(selector: (Char) ->
    Float): Float {
        if (isEmpty()) throw NoSuchElementException()
        var maxValue = selector(this[0])
        for
        (i in 1..lastIndex) {
            val v = selector(this[i])
            maxValue = maxOf(maxValue, v)
        }
        return
        maxValue
    }

```

```

maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each character in the char sequence.\n * \n * @throws NoSuchElementException if the char sequence is empty.\n *\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution\n ByLambdaReturnType\n @kotlin.internal.InlineOnly\n public inline fun <R : Comparable<R>>
CharSequence.maxOf(selector: (Char) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n
maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each character in the char sequence or `null` if there are no
characters.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution\n
ByLambdaReturnType\n @kotlin.internal.InlineOnly\n public inline fun CharSequence.maxOfOrNull(selector:
(Char) -> Double): Double? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each character in the char sequence or `null` if there are no characters.\n * \n * If any of values produced by
[selector] function is `NaN`, the returned result is `NaN`.\n
*\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution\n
ByLambdaReturnType\n @kotlin.internal.InlineOnly\n public inline fun CharSequence.maxOfOrNull(selector:
(Char) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each character in the char sequence or `null` if there are no characters.\n
*\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution\n
ByLambdaReturnType\n @kotlin.internal.InlineOnly\n public inline fun <R : Comparable<R>>
CharSequence.maxOfOrNull(selector: (Char) -> R): R? {\n if (isEmpty()) return null\n var maxValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n
maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each character in the char sequence.\n
*\n * \n * @throws NoSuchElementException if the char sequence is empty.\n
*\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution\n
ByLambdaReturnType\n @kotlin.internal.InlineOnly\n public inline fun <R> CharSequence.maxOfWith(comparator:
Comparator<in R>, selector: (Char) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each character in the char sequence or `null` if there are no characters.\n
*\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution\n
ByLambdaReturnType\n @kotlin.internal.InlineOnly\n public inline fun <R>
CharSequence.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {\n if (isEmpty())
return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest character or `null` if there are no characters.\n
*\n *\n @SinceKotlin("1.4")\n public fun
CharSequence.maxOrNull(): Char? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex)
{\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the first character
having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the
char sequence is empty.\n
*\n *\n @SinceKotlin("1.7")\n @kotlin.jvm.JvmName("maxWithOrThrow")\n @Suppress("CONFLICTING_OVER
LOADS")\n public fun CharSequence.maxWith(comparator: Comparator<in Char>): Char {\n if (isEmpty())

```

```

throw NoSuchElementException()\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if
(comparator.compare(max, e) < 0) max = e\n    }\n    return max\n}\n\n/**\n * Returns the first character having the
largest value according to the provided [comparator] or `null` if there are no characters.\n
*\n\n@SinceKotlin("1.4")\npublic fun CharSequence.maxWithOrNull(comparator: Comparator<in Char>): Char?
{\n    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if
(comparator.compare(max, e) < 0) max = e\n    }\n    return max\n}\n\n/**\n * Returns the smallest character.\n *
*\n * @throws NoSuchElementException if the char sequence is empty.\n
*\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow")\n@Suppress("CONFLICTING_OVERLOA
DS")\npublic fun CharSequence.min(): Char {\n    if (isEmpty()) throw NoSuchElementException()\n    var min =
this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return min\n}\n\n/**\n * Returns the first character yielding the smallest value of the given function.\n *
*\n * @throws
NoSuchElementException if the char sequence is empty.\n *
*\n * @sample
samples.collections.Collections.Aggregates.minBy\n
*\n\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minByOrThrow")\n@Suppress("CONFLICTING_OVERLO
ADS")\npublic inline fun <R : Comparable<R>> CharSequence.minBy(selector: (Char) -> R): Char {\n    if
(isEmpty()) throw NoSuchElementException()\n    var minElem = this[0]\n    val lastIndex = this.lastIndex\n    if
(lastIndex == 0) return minElem\n    var minValue = selector(minElem)\n    for (i in 1..lastIndex) {\n        val e =
this[i]\n        val v = selector(e)\n        if (minValue > v) {\n            minElem = e\n            minValue = v\n        }\n    }\n    return minElem\n}\n\n/**\n * Returns the first character yielding the smallest value of the given function or
`null` if there are no characters.\n *
*\n * @sample
samples.collections.Collections.Aggregates.minByOrNull\n
*\n\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> CharSequence.minByOrNull(selector: (Char) -
> R): Char? {\n    if (isEmpty()) return null\n    var minElem = this[0]\n    val lastIndex = this.lastIndex\n    if
(lastIndex == 0) return minElem\n    var minValue = selector(minElem)\n    for (i in 1..lastIndex) {\n        val e =
this[i]\n        val v = selector(e)\n        if (minValue > v) {\n            minElem = e\n            minValue = v\n        }\n    }\n    return minElem\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n *
*\n * applied to each character in the char sequence.\n *
*\n * If any of values produced by [selector] function is `NaN`, the
returned result is `NaN`.\n *
*\n * @throws NoSuchElementException if the char sequence is empty.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.minOf(selector: (Char) ->
Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for
(i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n *
*\n * applied to
each character in the char sequence.\n *
*\n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n *
*\n * @throws NoSuchElementException if the char sequence is empty.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.minOf(selector: (Char) ->
Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in
1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n *
*\n * applied to
each character in the char sequence.\n *
*\n * @throws NoSuchElementException if the char sequence is empty.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
CharSequence.minOf(selector: (Char) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var
minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n
            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values
produced by [selector] function\n *
*\n * applied to each character in the char sequence or `null` if there are no
characters.\n *
*\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

```



```

ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.minOrNull(selector:
(Char) -> Double): Double? {\n  if (isEmpty()) return null\n  var minValue = selector(this[0])\n  for (i in
1..lastIndex) {\n    val v = selector(this[i])\n    minValue = minOf(minValue, v)\n  }\n  return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each character in the char sequence or `null` if there are no characters.\n * \n * If any of values produced by
[selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.minOrNull(selector:
(Char) -> Float): Float? {\n  if (isEmpty()) return null\n  var minValue = selector(this[0])\n  for (i in
1..lastIndex) {\n    val v = selector(this[i])\n    minValue = minOf(minValue, v)\n  }\n  return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each character in the char sequence or `null` if there are no characters.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
CharSequence.minOrNull(selector: (Char) -> R): R? {\n  if (isEmpty()) return null\n  var minValue =
selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (minValue > v) {\n
minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each character in the char sequence.\n
*\n * @throws NoSuchElementException if the char sequence is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharSequence.minOfWith(comparator:
Comparator<in R>, selector: (Char) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var
minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(minValue, v) > 0) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each character in the char sequence or `null` if there are no characters.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
CharSequence.minOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {\n  if (isEmpty())
return null\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n
if (comparator.compare(minValue, v) > 0) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest character or `null` if there are no characters.\n
*\n@SinceKotlin("1.4")\npublic fun
CharSequence.minOrNull(): Char? {\n  if (isEmpty()) return null\n  var min = this[0]\n  for (i in 1..lastIndex) {\n
val e = this[i]\n  if (min > e) min = e\n  }\n  return min\n}\n\n/**\n * Returns the first character having the
smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the char
sequence is empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow")\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
CharSequence.minWith(comparator: Comparator<in Char>): Char {\n  if (isEmpty()) throw
NoSuchElementException()\n  var min = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if
(comparator.compare(min, e) > 0) min = e\n  }\n  return min\n}\n\n/**\n * Returns the first character having the
smallest value according to the provided [comparator] or `null` if there are no characters.\n
*\n@SinceKotlin("1.4")\npublic fun
CharSequence.minWithOrNull(comparator: Comparator<in Char>): Char? {\n  if (isEmpty()) return null\n  var
min = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if
(comparator.compare(min, e) > 0) min = e\n  }\n  return min\n}\n\n/**\n * Returns `true` if the char sequence
has no characters.\n * \n * @sample samples.collections.Collections.Aggregates.none\n *\npublic fun
CharSequence.none(): Boolean {\n  return isEmpty()\n}\n\n/**\n * Returns `true` if no characters match the given
[predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n *\npublic inline fun
CharSequence.none(predicate: (Char) -> Boolean): Boolean {\n  for (element in this) if (predicate(element)) return

```

```

false\n return true\n}\n\n/**\n * Performs the given [action] on each character and returns the char sequence itself afterwards.\n */\n@SinceKotlin("1.1")\npublic inline fun <S : CharSequence> S.onEach(action: (Char) -> Unit): S {\n return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each character, providing sequential index with the character,\n * and returns the char sequence itself afterwards.\n * @param [action] function that takes the index of a character and the character itself\n * and performs the action on the character.\n */\n@SinceKotlin("1.4")\npublic inline fun <S : CharSequence> S.onEachIndexed(action: (index: Int, Char) -> Unit): S {\n return apply { forEachIndexed(action) }\n}\n\n/**\n * Accumulates value starting with the first character and applying [operation] from left to right\n * to current accumulator value and each character.\n * \n * Throws an exception if this char sequence is empty. If the char sequence can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and a character,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\n\npublic inline fun CharSequence.reduce(operation: (acc: Char, Char) -> Char): Char {\n if (isEmpty())\n throw UnsupportedOperationException("Empty char sequence can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first character and applying [operation] from left to right\n * to current accumulator value and each character with its index in the original char sequence.\n * \n * Throws an exception if this char sequence is empty. If the char sequence can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of a character, current accumulator value and the character itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\n\npublic inline fun CharSequence.reduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): Char {\n if (isEmpty())\n throw UnsupportedOperationException("Empty char sequence can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first character and applying [operation] from left to right\n * to current accumulator value and each character with its index in the original char sequence.\n * \n * Returns `null` if the char sequence is empty.\n * \n * @param [operation] function that takes the index of a character, current accumulator value and the character itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun CharSequence.reduceIndexedOrNull(operation: (index: Int, acc: Char, Char) -> Char): Char? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first character and applying [operation] from left to right\n * to current accumulator value and each character.\n * \n * Returns `null` if the char sequence is empty.\n * \n * @param [operation] function that takes current accumulator value and a character,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun CharSequence.reduceOrNull(operation: (acc: Char, Char) -> Char): Char? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last character and applying [operation] from right to left\n * to each character and current accumulator value.\n * \n * Throws an exception if this char sequence is empty. If the char sequence can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes a character and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\n\npublic inline fun CharSequence.reduceRight(operation: (Char, acc: Char) -> Char): Char {\n var index = lastIndex\n if (index < 0)\n throw UnsupportedOperationException("Empty char sequence can't be reduced.")\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}

```

```

accumulator\n}\n/**\n * Accumulates value starting with the last character and applying [operation] from right to
left\n * to each character with its index in the original char sequence and current accumulator value.\n * \n * Throws
an exception if this char sequence is empty. If the char sequence can be empty in an expected way,\n * please use
[reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation]
function that takes the index of a character, the character itself and current accumulator value,\n * and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic
inline fun CharSequence.reduceRightIndexed(operation: (index: Int, Char, acc: Char) -> Char): Char {\n    var index
= lastIndex\n    if (index < 0) throw UnsupportedOperationException("Empty char sequence can't be reduced.")\n
    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index),
accumulator)\n        --index\n    }\n    return accumulator\n}\n}\n/**\n * Accumulates value starting with the last
character and applying [operation] from right to left\n * to each character with its index in the original char sequence
and current accumulator value.\n * \n * Returns `null` if the char sequence is empty.\n * \n * @param [operation]
function that takes the index of a character, the character itself and current accumulator value,\n * and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n
*/\n@SinceKotlin("1.4")\npublic inline fun CharSequence.reduceRightIndexedOrNull(operation: (index: Int, Char,
acc: Char) -> Char): Char? {\n    var index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index-
-)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n        --index\n    }\n
return accumulator\n}\n}\n/**\n * Accumulates value starting with the last character and applying [operation] from
right to left\n * to each character and current accumulator value.\n * \n * Returns `null` if the char sequence is
empty.\n * \n * @param [operation] function that takes a character and current accumulator value,\n * and calculates
the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
CharSequence.reduceRightOrNull(operation: (Char, acc: Char) -> Char): Char? {\n    var index = lastIndex\n    if
(index < 0) return null\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator =
operation(get(index--), accumulator)\n    }\n    return accumulator\n}\n}\n/**\n * Returns a list containing successive
accumulation values generated by applying [operation] from left to right\n * to each character and current
accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should
not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes current accumulator value and a character, and calculates the next accumulator value.\n * \n *
@sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin("1.4")\npublic inline fun
<R> CharSequence.runningFold(initial: R, operation: (acc: R, Char) -> R): List<R> {\n    if (isEmpty()) return
listOf(initial)\n    val result = ArrayList<R>(length + 1).apply { add(initial) }\n    var accumulator = initial\n
for (element in this) {\n        accumulator = operation(accumulator, element)\n        result.add(accumulator)\n    }\n
return result\n}\n}\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each character, its index in the original char sequence and current accumulator
value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be
mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that
takes the index of a character, current accumulator value\n * and the character itself, and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*/\n@SinceKotlin("1.4")\npublic inline fun <R> CharSequence.runningFoldIndexed(initial: R, operation: (index:
Int, acc: R, Char) -> R): List<R> {\n    if (isEmpty()) return listOf(initial)\n    val result = ArrayList<R>(length +
1).apply { add(initial) }\n    var accumulator = initial\n    for (index in indices) {\n        accumulator =
operation(index, accumulator, this[index])\n        result.add(accumulator)\n    }\n    return result\n}\n}\n/**\n *
Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to
each character and current accumulator value that starts with the first character of this char sequence.\n * \n * Note
that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous
value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and a character,
and calculates the next accumulator value.\n * \n * @sample

```

```

samples.collections.Collections.Aggregates.runningReduce\n *^n@SinceKotlin("1.4")\npublic inline fun
CharSequence.runningReduce(operation: (acc: Char, Char) -> Char): List<Char> {\n  if (isEmpty()) return
emptyList()\n  var accumulator = this[0]\n  val result = ArrayList<Char>(length).apply { add(accumulator) }\n
for (index in 1 until length) {\n    accumulator = operation(accumulator, this[index])\n
result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each character, its index in the original char
sequence and current accumulator value that starts with the first character of this char sequence.\n * \n * Note that
`acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in
resulting list.\n * \n * @param [operation] function that takes the index of a character, current accumulator value\n *
and the character itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n *^n@SinceKotlin("1.4")\npublic inline fun
CharSequence.runningReduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): List<Char> {\n  if
(isEmpty()) return emptyList()\n  var accumulator = this[0]\n  val result = ArrayList<Char>(length).apply {
add(accumulator) }\n  for (index in 1 until length) {\n    accumulator = operation(index, accumulator,
this[index])\n    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list containing successive
accumulation values generated by applying [operation] from left to right\n * to each character and current
accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should
not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes current accumulator value and a character, and calculates the next accumulator value.\n * \n *
@sample samples.collections.Collections.Aggregates.scan\n
*^n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <R>
CharSequence.scan(initial: R, operation: (acc: R, Char) -> R): List<R> {\n  return runningFold(initial,
operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation]
from left to right\n * to each character, its index in the original char sequence and current accumulator value that
starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the
index of a character, current accumulator value\n * and the character itself, and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*^n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <R>
CharSequence.scanIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): List<R> {\n  return
runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each character in the char sequence.\n * \n * @Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
CharSequence.sumBy(selector: (Char) -> Int): Int {\n  var sum: Int = 0\n  for (element in this) {\n    sum +=
selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each character in the char sequence.\n * \n * @Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
CharSequence.sumByDouble(selector: (Char) -> Double): Double {\n  var sum: Double = 0.0\n  for (element in
this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each character in the char sequence.\n
*^n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
CharSequence.sumOf(selector: (Char) -> Double): Double {\n  var sum: Double = 0.toDouble()\n  for (element in
this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each character in the char sequence.\n
*^n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun
CharSequence.sumOf(selector: (Char) -> Int): Int {\n  var sum: Int = 0.toInt()\n  for (element in this) {\n    sum

```

```

+= selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each character in the char sequence.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun
CharSequence.sumOf(selector: (Char) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this) {\n
        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each character in the char sequence.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.sumOf(selector: (Char) -> UInt): UInt {\n
    var sum: UInt = 0.toUInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each character in the char
sequence.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.sumOf(selector: (Char) -> ULong):
ULong {\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n
return sum\n}\n\n/**\n * Splits this char sequence into a list of strings each not exceeding the given [size].\n * \n *
The last string in the resulting list may have fewer characters than the given [size].\n * \n * @param size the number
of elements to take in each string, must be positive and can be greater than the number of elements in this char
sequence.\n * \n * @sample samples.text.Strings.chunked\n *\n@SinceKotlin("1.2")\npublic fun
CharSequence.chunked(size: Int): List<String> {\n    return windowed(size, size, partialWindows =
true)\n}\n\n/**\n * Splits this char sequence into several char sequences each not exceeding the given [size]\n * and
applies the given [transform] function to an each.\n * \n * @return list of results of the [transform] applied to an
each char sequence.\n * \n * Note that the char sequence passed to the [transform] function is ephemeral and is valid
only inside that function.\n * You should not store it or allow it to escape in some way, unless you made a snapshot
of it.\n * The last char sequence may have fewer characters than the given [size].\n * \n * @param size the number
of elements to take in each char sequence, must be positive and can be greater than the number of elements in this
char sequence.\n * \n * @sample samples.text.Strings.chunkedTransform\n *\n@SinceKotlin("1.2")\npublic fun
<R> CharSequence.chunked(size: Int, transform: (CharSequence) -> R): List<R> {\n    return windowed(size, size,
partialWindows = true, transform = transform)\n}\n\n/**\n * Splits this char sequence into a sequence of strings
each not exceeding the given [size].\n * \n * The last string in the resulting sequence may have fewer characters than
the given [size].\n * \n * @param size the number of elements to take in each string, must be positive and can be
greater than the number of elements in this char sequence.\n * \n * @sample
samples.collections.Collections.Transformations.chunked\n *\n@SinceKotlin("1.2")\npublic fun
CharSequence.chunkedSequence(size: Int): Sequence<String> {\n    return chunkedSequence(size) { it.toString()
}\n}\n\n/**\n * Splits this char sequence into several char sequences each not exceeding the given [size]\n * and
applies the given [transform] function to an each.\n * \n * @return sequence of results of the [transform] applied to
an each char sequence.\n * \n * Note that the char sequence passed to the [transform] function is ephemeral and is
valid only inside that function.\n * You should not store it or allow it to escape in some way, unless you made a
snapshot of it.\n * The last char sequence may have fewer characters than the given [size].\n * \n * @param size the
number of elements to take in each char sequence, must be positive and can be greater than the number of elements
in this char sequence.\n * \n * @sample samples.text.Strings.chunkedTransformToSequence\n
*\n@SinceKotlin("1.2")\npublic fun <R> CharSequence.chunkedSequence(size: Int, transform: (CharSequence) -
> R): Sequence<R> {\n    return windowedSequence(size, size, partialWindows = true, transform =
transform)\n}\n\n/**\n * Splits the original char sequence into pair of char sequences,\n * where *first* char
sequence contains characters for which [predicate] yielded `true`,\n * while *second* char sequence contains
characters for which [predicate] yielded `false`.\n * \n * @sample samples.text.Strings.partition\n *\npublic inline

```

```

fun CharSequence.partition(predicate: (Char) -> Boolean): Pair<CharSequence, CharSequence> {
    val first = StringBuilder()
    val second = StringBuilder()
    for (element in this) {
        if (predicate(element)) {
            first.append(element)
        } else {
            second.append(element)
        }
    }
    return Pair(first, second)
}

// Splits the original string into pair of strings, where *first* string contains characters for
// which [predicate] yielded `true`, while *second* string contains characters for which [predicate] yielded
// `false`.
// @sample samples.text.Strings.partition
// public inline fun String.partition(predicate: (Char) -> Boolean): Pair<String, String> {
//     val first = StringBuilder()
//     val second = StringBuilder()
//     for (element in this) {
//         if (predicate(element)) {
//             first.append(element)
//         } else {
//             second.append(element)
//         }
//     }
//     return Pair(first.toString(), second.toString())
// }

// Returns a list of snapshots of the window of the given [size] sliding along this char sequence with the given [step], where
// each snapshot is a string. Several last strings may have fewer characters than the given [size].
// Both [size] and [step] must be positive and can be greater than the number of elements in this char sequence.
// @param size the number of elements to take in each window
// @param step the number of elements to move the window forward by on an each step, by default 1
// @param partialWindows controls whether or not to keep partial windows in the end if any, by default `false` which means partial windows won't be preserved
// @sample samples.collections.Sequences.Transformations.takeWindows
// @SinceKotlin("1.2")
// public fun CharSequence.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false): List<String> {
//     return windowed(size, step, partialWindows) { it.toString() }
// }

// Returns a list of results of applying the given [transform] function to an each char sequence representing a view over the window of the given [size] sliding
// along this char sequence with the given [step]. Note that the char sequence passed to the [transform] function is ephemeral and is valid only inside that function.
// You should not store it or allow it to escape in some way, unless you made a snapshot of it. Several last char sequences may have fewer characters than the given
// [size]. Both [size] and [step] must be positive and can be greater than the number of elements in this char sequence.
// @param size the number of elements to take in each window
// @param step the number of elements to move the window forward by on an each step, by default 1
// @param partialWindows controls whether or not to keep partial windows in the end if any, by default `false` which means partial windows won't
// be preserved
// @sample samples.collections.Sequences.Transformations.averageWindows
// @SinceKotlin("1.2")
// public fun <R> CharSequence.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (CharSequence) -> R): List<R> {
//     checkWindowSizeStep(size, step)
//     val thisSize = this.length
//     val resultCapacity = thisSize / step + if (thisSize % step == 0) 0 else 1
//     val result = ArrayList<R>(resultCapacity)
//     var index = 0
//     while (index in 0 until thisSize) {
//         val end = index + size
//         val coercedEnd = if (end < 0 || end > thisSize) { if (partialWindows) thisSize else break } else end
//         result.add(transform(subSequence(index, coercedEnd)))
//         index += step
//     }
//     return result
// }

// Returns a sequence of snapshots of the window of the given [size] sliding along this char sequence with the given [step], where each snapshot is a string.
// Several last strings may have fewer characters than the given [size]. Both [size] and [step] must be positive and can be greater than the number of elements in this
// char sequence. @param size the number of elements to take in each window @param step the number of elements to move the window forward by on an each step, by default 1
// @param partialWindows controls whether or not to keep partial windows in the end if any, by default `false` which means partial windows won't be preserved
// @sample samples.collections.Sequences.Transformations.takeWindows
// @SinceKotlin("1.2")
// public fun CharSequence.windowedSequence(size: Int, step: Int = 1, partialWindows: Boolean = false): Sequence<String> {
//     return windowedSequence(size, step, partialWindows) { it.toString() }
// }

// Returns a sequence of results of applying the given [transform] function to an each char sequence representing a view over the window of the given [size] sliding
// along this char sequence with the given [step]. Note that the char sequence passed to the [transform] function is ephemeral and is valid only inside
// that function. You should not store it or allow it to escape in some way, unless you made a snapshot of it. Several last char sequences may have fewer characters than the given
// [size]. Both [size] and [step] must be positive and can be greater than the number of elements in this char sequence. @param size the number of

```

elements to take in each window\n * @param step the number of elements to move the window forward by on each step, by default 1\n * @param partialWindows controls whether or not to keep partial windows in the end if any,\n * by default `false` which means partial windows won't be preserved\n * \n * @sample samples.collections.Sequences.Transformations.averageWindows\n * \n * @SinceKotlin("1.2")\n * \n * public fun <R> CharSequence.windowedSequence(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (CharSequence) -> R): Sequence<R> {\n * checkWindowSizeStep(size, step)\n * val windows = (if (partialWindows) indices else 0 until length - size + 1) step step\n * return windows.asSequence().map { index ->\n * val end = index + size\n * val coercedEnd = if (end < 0 || end > length) length else end\n * transform(subSequence(index, coercedEnd))\n * }\n * }\n * \n * Returns a list of pairs built from the characters of `this` and the [other] char sequences with the same index\n * The returned list has length of the shortest char sequence.\n * \n * @sample samples.text.Strings.zip\n * \n * public infix fun CharSequence.zip(other: CharSequence): List<Pair<Char, Char>> {\n * return zip(other) { c1, c2 -> c1 to c2 }\n * }\n * \n * Returns a list of values built from the characters of `this` and the [other] char sequences with the same index\n * using the provided [transform] function applied to each pair of characters.\n * The returned list has length of the shortest char sequence.\n * \n * @sample samples.text.Strings.zipWithTransform\n * \n * public inline fun <V> CharSequence.zip(other: CharSequence, transform: (a: Char, b: Char) -> V): List<V> {\n * val length = minOf(this.length, other.length)\n * val list = ArrayList<V>(length)\n * for (i in 0 until length) {\n * list.add(transform(this[i], other[i]))\n * }\n * return list\n * }\n * \n * Returns a list of pairs of each two adjacent characters in this char sequence.\n * The returned list is empty if this char sequence contains less than two characters.\n * \n * @sample samples.collections.Collections.Transformations.zipWithNext\n * \n * @SinceKotlin("1.2")\n * \n * public fun CharSequence.zipWithNext(): List<Pair<Char, Char>> {\n * return zipWithNext { a, b -> a to b }\n * }\n * \n * Returns a list containing the results of applying the given [transform] function\n * to an each pair of two adjacent characters in this char sequence.\n * The returned list is empty if this char sequence contains less than two characters.\n * \n * @sample samples.collections.Collections.Transformations.zipWithNextToFindDeltas\n * \n * @SinceKotlin("1.2")\n * \n * public inline fun <R> CharSequence.zipWithNext(transform: (a: Char, b: Char) -> R): List<R> {\n * val size = length - 1\n * if (size < 1) return emptyList()\n * val result = ArrayList<R>(size)\n * for (index in 0 until size) {\n * result.add(transform(this[index], this[index + 1]))\n * }\n * return result\n * }\n * \n * Creates an [Iterable] instance that wraps the original char sequence returning its characters when being iterated.\n * \n * \n * public fun CharSequence.asIterable(): Iterable<Char> {\n * if (this is String && isEmpty()) return emptyList()\n * return Iterable { this.iterator() }\n * }\n * \n * Creates a [Sequence] instance that wraps the original char sequence returning its characters when being iterated.\n * \n * \n * public fun CharSequence.asSequence(): Sequence<Char> {\n * if (this is String && isEmpty()) return emptySequence()\n * return Sequence { this.iterator() }\n * }\n * \n * \n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n * \n * \n * @file:kotlin.jvm.JvmMultifileClass\n * @file:kotlin.jvm.JvmName("StringsKt")\n * \n * package kotlin.text\n * \n * import kotlin.contracts.contract\n * import kotlin.jvm.JvmName\n * \n * Returns a copy of this string converted to upper case using the rules of the default locale.\n * \n * @Deprecated("Use uppercase() instead.")\n * ReplaceWith("uppercase()")\n * \n * @DeprecatedSinceKotlin(warningSince = "1.5")\n * \n * public expect fun String.toUpperCase(): String\n * \n * Returns a copy of this string converted to upper case using Unicode mapping rules of the invariant locale.\n * \n * This function supports one-to-many and many-to-one character mapping,\n * thus the length of the returned string can be different from the length of the original string.\n * \n * @sample samples.text.Strings.toUpperCase\n * \n * \n * @SinceKotlin("1.5")\n * @WasExperimental(ExperimentalStdlibApi::class)\n * \n * public expect fun String.toUpperCase(): String\n * \n * Returns a copy of this string converted to lower case using the rules of the default locale.\n * \n * @Deprecated("Use lowercase() instead.")\n * ReplaceWith("lowercase()")\n * \n * @DeprecatedSinceKotlin(warningSince = "1.5")\n * \n * public expect fun String.toLowerCase(): String\n * \n * Returns a copy of this string converted to lower case using Unicode mapping rules of the invariant locale.\n * \n * This function supports one-to-many and many-to-one character

```

mapping,\n * thus the length of the returned string can be different from the length of the original string.\n *\n *
@sample samples.text.Strings.lowercase\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
String.lowercase(): String\n/>\n * Returns a copy of this string having its first letter titlecased using the rules of
the default locale,\n * or the original string if it's empty or already starts with a title case letter.\n *\n * The title case
of a character is usually the same as its upper case with several exceptions.\n * The particular list of characters with
the special title case form depends on the underlying platform.\n *\n * @sample samples.text.Strings.capitalize\n
*\n@Deprecated("Use replaceFirstChar instead.", ReplaceWith("replaceFirstChar { if (it.isLowerCase())
it.titlecase() else it.toString() }"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic expect fun
String.capitalize(): String\n/>\n * Returns a copy of this string having its first letter lowercased using the rules of
the default locale,\n * or the original string if it's empty or already starts with a lower case letter.\n *\n * @sample
samples.text.Strings.decapitalize\n
*\n@Deprecated("Use replaceFirstChar instead.",
ReplaceWith("replaceFirstChar { it.lowercase() }"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic
expect fun String.decapitalize(): String\n/>\n * Returns a sub sequence of this char sequence having leading and
trailing characters matching the [predicate] removed.\n *\npublic inline fun CharSequence.trim(predicate: (Char) ->
Boolean): CharSequence {\n    var startIndex = 0\n    var endIndex = length - 1\n    var startFound = false\n\n    while (startIndex <= endIndex) {\n        val index = if (!startFound) startIndex else endIndex\n        val match =
predicate(this[index])\n        if (!startFound) {\n            if (!match)\n                startFound = true\n            else\n                startIndex += 1\n        } else {\n            if (!match)\n                break\n            else\n                endIndex -= 1\n        }\n    }\n\n    return subSequence(startIndex, endIndex + 1)\n}\n/>\n * Returns a string having leading and
trailing characters matching the [predicate] removed.\n *\npublic inline fun String.trim(predicate: (Char) ->
Boolean): String =\n    (this as CharSequence).trim(predicate).toString()\n/>\n * Returns a sub sequence of this
char sequence having leading characters matching the [predicate] removed.\n *\npublic inline fun
CharSequence.trimStart(predicate: (Char) -> Boolean): CharSequence {\n    for (index in this.indices)\n        if
(!predicate(this[index]))\n            return subSequence(index, length)\n\n    return ""\n}\n/>\n * Returns a string
having leading characters matching the [predicate] removed.\n *\npublic inline fun String.trimStart(predicate:
(Char) -> Boolean): String =\n    (this as CharSequence).trimStart(predicate).toString()\n/>\n * Returns a sub
sequence of this char sequence having trailing characters matching the [predicate] removed.\n *\npublic inline fun
CharSequence.trimEnd(predicate: (Char) -> Boolean): CharSequence {\n    for (index in this.indices.reversed())\n        if
(!predicate(this[index]))\n            return subSequence(0, index + 1)\n\n    return ""\n}\n/>\n * Returns a string
having trailing characters matching the [predicate] removed.\n *\npublic inline fun String.trimEnd(predicate: (Char)
-> Boolean): String =\n    (this as CharSequence).trimEnd(predicate).toString()\n/>\n * Returns a sub sequence of
this char sequence having leading and trailing characters from the [chars] array removed.\n *\npublic fun
CharSequence.trim(vararg chars: Char): CharSequence = trim { it in chars }\n/>\n * Returns a string having
leading and trailing characters from the [chars] array removed.\n *\npublic fun String.trim(vararg chars: Char):
String = trim { it in chars }\n/>\n * Returns a sub sequence of this char sequence having leading characters from
the [chars] array removed.\n *\npublic fun CharSequence.trimStart(vararg chars: Char): CharSequence = trimStart {
it in chars }\n/>\n * Returns a string having leading characters from the [chars] array removed.\n *\npublic fun
String.trimStart(vararg chars: Char): String = trimStart { it in chars }\n/>\n * Returns a sub sequence of this char
sequence having trailing characters from the [chars] array removed.\n *\npublic fun CharSequence.trimEnd(vararg
chars: Char): CharSequence = trimEnd { it in chars }\n/>\n * Returns a string having trailing characters from the
[chars] array removed.\n *\npublic fun String.trimEnd(vararg chars: Char): String = trimEnd { it in chars }\n/>\n
* Returns a sub sequence of this char sequence having leading and trailing whitespace removed.\n *\npublic fun
CharSequence.trim(): CharSequence = trim(Char::isWhitespace)\n/>\n * Returns a string having leading and
trailing whitespace removed.\n *\n@kotlin.internal.InlineOnly\npublic inline fun String.trim(): String = (this as
CharSequence).trim().toString()\n/>\n * Returns a sub sequence of this char sequence having leading whitespace
removed.\n *\npublic fun CharSequence.trimStart(): CharSequence = trimStart(Char::isWhitespace)\n/>\n *
Returns a string having leading whitespace removed.\n *\n@kotlin.internal.InlineOnly\npublic inline fun

```



```

String.trimStart(): String = (this as CharSequence).trimStart().toString()\n\n/**\n * Returns a sub sequence of this
char sequence having trailing whitespace removed.\n */\npublic fun CharSequence.trimEnd(): CharSequence =
trimEnd(Char::isWhitespace)\n\n/**\n * Returns a string having trailing whitespace removed.\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun String.trimEnd(): String = (this as
CharSequence).trimEnd().toString()\n\n/**\n * Returns a char sequence with content of this char sequence padded at
the beginning\n * to the specified [length] with the specified character or space.\n */\n * @param length the desired
string length.\n * @param padChar the character to pad string with, if it has length less than the [length] specified.
Space is used by default.\n * @return Returns a char sequence of length at least [length] consisting of `this` char
sequence prepended with [padChar] as many times\n * as are necessary to reach that length.\n * @sample
samples.text.Strings.padStart\n */\npublic fun CharSequence.padStart(length: Int, padChar: Char = ' '):
CharSequence {\n    if (length < 0)\n        throw IllegalArgumentException("Desired length $length is less than
zero.")\n    if (length <= this.length)\n        return this.subSequence(0, this.length)\n\n    val sb =
StringBuilder(length)\n    for (i in 1..(length - this.length))\n        sb.append(padChar)\n    sb.append(this)\n    return
sb\n}\n\n/**\n * Pads the string to the specified [length] at the beginning with the specified character or space.\n */\n *
@param length the desired string length.\n * @param padChar the character to pad string with, if it has length less
than the [length] specified. Space is used by default.\n * @return Returns a string of length at least [length]
consisting of `this` string prepended with [padChar] as many times\n * as are necessary to reach that length.\n *
@sample samples.text.Strings.padStart\n */\npublic fun String.padStart(length: Int, padChar: Char = ' '): String =\n(this as CharSequence).padStart(length, padChar).toString()\n\n/**\n * Returns a char sequence with content of this
char sequence padded at the end\n * to the specified [length] with the specified character or space.\n */\n * @param
length the desired string length.\n * @param padChar the character to pad string with, if it has length less than the
[length] specified. Space is used by default.\n * @return Returns a char sequence of length at least [length]
consisting of `this` char sequence appended with [padChar] as many times\n * as are necessary to reach that
length.\n * @sample samples.text.Strings.padEnd\n */\npublic fun CharSequence.padEnd(length: Int, padChar: Char
= ' '): CharSequence {\n    if (length < 0)\n        throw IllegalArgumentException("Desired length $length is less
than zero.")\n    if (length <= this.length)\n        return this.subSequence(0, this.length)\n\n    val sb =
StringBuilder(length)\n    sb.append(this)\n    for (i in 1..(length - this.length))\n        sb.append(padChar)\n    return
sb\n}\n\n/**\n * Pads the string to the specified [length] at the end with the specified character or space.\n */\n *
@param length the desired string length.\n * @param padChar the character to pad string with, if it has length less
than the [length] specified. Space is used by default.\n * @return Returns a string of length at least [length]
consisting of `this` string appended with [padChar] as many times\n * as are necessary to reach that length.\n *
@sample samples.text.Strings.padEnd\n */\npublic fun String.padEnd(length: Int, padChar: Char = ' '): String =\n(this as CharSequence).padEnd(length, padChar).toString()\n\n/**\n * Returns `true` if this nullable char sequence is
either `null` or empty.\n */\n * @sample samples.text.Strings.stringOrNullEmpty\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence?.isNullOrEmpty(): Boolean {\n    contract {\n
returns(false) implies (this@isNullOrEmpty != null)\n    }\n\n    return this == null || this.length == 0\n}\n\n/**\n *
Returns `true` if this char sequence is empty (contains no characters).\n */\n * @sample
samples.text.Strings.stringIsEmpty\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.isEmpty():
Boolean = length == 0\n\n/**\n * Returns `true` if this char sequence is not empty.\n */\n * @sample
samples.text.Strings.stringIsNotEmpty\n */\n@kotlin.internal.InlineOnly\npublic inline fun
CharSequence.isNotEmpty(): Boolean = length > 0\n\n// implemented differently in JVM and JS\n//\npublic fun
String.isBlank(): Boolean = length() == 0 || all { it.isWhitespace() }\n\n/**\n * Returns `true` if this char sequence
is not empty and contains some characters except of whitespace characters.\n */\n * @sample
samples.text.Strings.stringIsNotBlank\n */\n@kotlin.internal.InlineOnly\npublic inline fun
CharSequence.isNotBlank(): Boolean = !isBlank()\n\n/**\n * Returns `true` if this nullable char sequence is either
`null` or empty or consists solely of whitespace characters.\n */\n * @sample
samples.text.Strings.stringOrNullBlank\n */\n@kotlin.internal.InlineOnly\npublic inline fun
CharSequence?.isNullOrBlank(): Boolean {\n    contract {\n        returns(false) implies (this@isNullOrBlank !=

```

```

null)\n } \n\n return this == null || this.isBlank()\n}\n\n/**\n * Iterator for characters of the given char sequence.\n *\n * public operator fun CharSequence.iterator(): CharIterator = object : CharIterator() {\n *     private var index = 0\n *     public override fun nextChar(): Char = get(index++)\n *     public override fun hasNext(): Boolean = index < length\n * }\n\n/**\n * Returns the string if it is not `null`, or the empty string otherwise.\n *\n * @kotlin.internal.InlineOnly\n * public inline fun String?.orEmpty(): String = this ?: ""\n\n/**\n * Returns this char sequence if it's not empty\n * or the result of calling [defaultValue] function if the char sequence is empty.\n *\n * @sample samples.text.Strings.stringIfEmpty\n\n * @kotlin("1.3")\n * @kotlin.internal.InlineOnly\n * public inline fun <C, R> C.ifEmpty(defaultValue: () -> R): R where C : CharSequence, C : R =\n *     if (isEmpty()) defaultValue() else this\n\n/**\n * Returns this char sequence if it is not empty and doesn't consist solely of whitespace characters,\n * or the result of calling [defaultValue] function otherwise.\n *\n * @sample samples.text.Strings.stringIfBlank\n\n * @kotlin("1.3")\n * @kotlin.internal.InlineOnly\n * public inline fun <C, R> C.ifBlank(defaultValue: () -> R): R where C : CharSequence, C : R =\n *     if (isBlank()) defaultValue() else this\n\n/**\n * Returns the range of valid character indices for this char sequence.\n *\n * public val CharSequence.indices: IntRange\n *     get() = 0..length - 1\n\n/**\n * Returns the index of the last character in the char sequence or -1 if it is empty.\n *\n * public val CharSequence.lastIndex: Int\n *     get() = this.length - 1\n\n/**\n * Returns `true` if this CharSequence has Unicode surrogate pair at the specified [index].\n *\n * public fun CharSequence.hasSurrogatePairAt(index: Int): Boolean {\n *     return index in 0..length - 2\n *         && this[index].isHighSurrogate()\n *         && this[index + 1].isLowSurrogate()\n * }\n\n/**\n * Returns a substring specified by the given [range] of indices.\n *\n * public fun String.substring(range: IntRange): String = substring(range.start, range.endInclusive + 1)\n\n/**\n * Returns a subsequence of this char sequence specified by the given [range] of indices.\n *\n * public fun CharSequence.subSequence(range: IntRange): CharSequence = subSequence(range.start, range.endInclusive + 1)\n\n/**\n * Returns a subsequence of this char sequence.\n *\n * This extension is chosen only for invocation with old-named parameters.\n * Replace parameter names with the same as those of [CharSequence.subSequence].\n\n * @kotlin.internal.InlineOnly\n * @Suppress("EXTENSION_SHADOWED_BY_MEMBER") // false\n * warning\n * @Deprecated("Use parameters named startIndex and endIndex.", ReplaceWith("subSequence(startIndex = start, endIndex = end)"))\n * public inline fun String.subSequence(start: Int, end: Int): CharSequence = subSequence(start, end)\n\n/**\n * Returns a substring of chars from a range of this char sequence starting at the [startIndex] and ending right before the [endIndex].\n *\n * @param startIndex the start index (inclusive).\n * @param endIndex the end index (exclusive). If not specified, the length of the char sequence is used.\n\n * @kotlin.internal.InlineOnly\n * public inline fun CharSequence.substring(startIndex: Int, endIndex: Int = length): String = subSequence(startIndex, endIndex).toString()\n\n/**\n * Returns a substring of chars at indices from the specified [range] of this char sequence.\n *\n * public fun CharSequence.substring(range: IntRange): String = subSequence(range.start, range.endInclusive + 1).toString()\n\n/**\n * Returns a substring before the first occurrence of [delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n *\n * public fun String.substringBefore(delimiter: Char, missingDelimiterValue: String = this): String {\n *     val index = indexOf(delimiter)\n *     return if (index == -1) missingDelimiterValue else substring(0, index)\n * }\n\n/**\n * Returns a substring before the first occurrence of [delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n *\n * public fun String.substringBefore(delimiter: String, missingDelimiterValue: String = this): String {\n *     val index = indexOf(delimiter)\n *     return if (index == -1) missingDelimiterValue else substring(0, index)\n * }\n\n/**\n * Returns a substring after the first occurrence of [delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n *\n * public fun String.substringAfter(delimiter: Char, missingDelimiterValue: String = this): String {\n *     val index = indexOf(delimiter)\n *     return if (index == -1) missingDelimiterValue else substring(index + 1, length)\n * }\n\n/**\n * Returns a substring after the first occurrence of [delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n *\n * public fun String.substringAfter(delimiter: String, missingDelimiterValue: String = this): String {\n *     val index = indexOf(delimiter)\n *     return if (index == -1) missingDelimiterValue else substring(index +

```

```

delimiter.length, length)\n}\n\n/**\n * Returns a substring before the last occurrence of [delimiter].\n * If the string
does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic
fun String.substringBeforeLast(delimiter: Char, missingDelimiterValue: String = this): String {\n    val index =
lastIndexOf(delimiter)\n    return if (index == -1) missingDelimiterValue else substring(0, index)\n}\n\n/**\n *
Returns a substring before the last occurrence of [delimiter].\n * If the string does not contain the delimiter, returns
[missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.substringBeforeLast(delimiter:
String, missingDelimiterValue: String = this): String {\n    val index = lastIndexOf(delimiter)\n    return if (index ==
-1) missingDelimiterValue else substring(0, index)\n}\n\n/**\n * Returns a substring after the last occurrence of
[delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the
original string.\n */\npublic fun String.substringAfterLast(delimiter: Char, missingDelimiterValue: String = this):
String {\n    val index = lastIndexOf(delimiter)\n    return if (index == -1) missingDelimiterValue else
substring(index + 1, length)\n}\n\n/**\n * Returns a substring after the last occurrence of [delimiter].\n * If the
string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\n
public fun String.substringAfterLast(delimiter: String, missingDelimiterValue: String = this): String {\n    val
index = lastIndexOf(delimiter)\n    return if (index == -1) missingDelimiterValue else substring(index +
delimiter.length, length)\n}\n\n/**\n * Returns a char sequence with content of this char sequence where its part at
the given range\n * is replaced with the [replacement] char sequence.\n * @param startIndex the index of the first
character to be replaced.\n * @param endIndex the index of the first character after the replacement to keep in the
string.\n */\npublic fun CharSequence.replaceRange(startIndex: Int, endIndex: Int, replacement: CharSequence):
CharSequence {\n    if (endIndex < startIndex)\n        throw IndexOutOfBoundsException("End index ($endIndex)
is less than start index ($startIndex).")\n    val sb = StringBuilder()\n    sb.appendRange(this, 0, startIndex)\n
sb.append(replacement)\n    sb.appendRange(this, endIndex, length)\n    return sb\n}\n\n/**\n * Replaces the part of
the string at the given range with the [replacement] char sequence.\n * @param startIndex the index of the first
character to be replaced.\n * @param endIndex the index of the first character after the replacement to keep in the
string.\n */\n@kotlin.internal.InlineOnly\npublic inline fun String.replaceRange(startIndex: Int, endIndex: Int,
replacement: CharSequence): String =\n    (this as CharSequence).replaceRange(startIndex, endIndex,
replacement).toString()\n\n/**\n * Returns a char sequence with content of this char sequence where its part at the
given [range]\n * is replaced with the [replacement] char sequence.\n * * The end index of the [range] is included
in the part to be replaced.\n */\npublic fun CharSequence.replaceRange(range: IntRange, replacement:
CharSequence): CharSequence =\n    replaceRange(range.start, range.endInclusive + 1, replacement)\n\n/**\n *
Replace the part of string at the given [range] with the [replacement] string.\n * * The end index of the [range] is
included in the part to be replaced.\n */\n@kotlin.internal.InlineOnly\npublic inline fun String.replaceRange(range:
IntRange, replacement: CharSequence): String =\n    (this as CharSequence).replaceRange(range,
replacement).toString()\n\n/**\n * Returns a char sequence with content of this char sequence where its part at the
given range is removed.\n * * @param startIndex the index of the first character to be removed.\n * * @param
endIndex the index of the first character after the removed part to keep in the string.\n * * [endIndex] is not
included in the removed part.\n */\npublic fun CharSequence.removeRange(startIndex: Int, endIndex: Int):
CharSequence {\n    if (endIndex < startIndex)\n        throw IndexOutOfBoundsException("End index ($endIndex)
is less than start index ($startIndex).")\n    if (endIndex == startIndex)\n        return this.subSequence(0,
length)\n    val sb = StringBuilder(length - (endIndex - startIndex))\n    sb.appendRange(this, 0, startIndex)\n
sb.appendRange(this, endIndex, length)\n    return sb\n}\n\n/**\n * Removes the part of a string at a given range.\n *
@param startIndex the index of the first character to be removed.\n * @param endIndex the index of the first
character after the removed part to keep in the string.\n * * [endIndex] is not included in the removed part.\n */\n
@kotlin.internal.InlineOnly\npublic inline fun String.removeRange(startIndex: Int, endIndex: Int): String =\n    (this as CharSequence).removeRange(startIndex, endIndex).toString()\n\n/**\n * Returns a char sequence with
content of this char sequence where its part at the given [range] is removed.\n * * The end index of the [range] is
included in the removed part.\n */\npublic fun CharSequence.removeRange(range: IntRange): CharSequence =
removeRange(range.start, range.endInclusive + 1)\n\n/**\n * Removes the part of a string at the given [range].\n *

```

* The end index of the [range] is included in the removed part.
`String.removeRange(range: IntRange): String` (this as `CharSequence`).
 * If this char sequence starts with the given [prefix], returns a new char sequence with the prefix removed.
 Otherwise, returns a new char sequence with the same characters.

`CharSequence.removePrefix(prefix: CharSequence): CharSequence`
 * If this string starts with the given [prefix], returns a copy of this string with the prefix removed. Otherwise, returns this string.

`String.removePrefix(prefix: CharSequence): String`
 * If this char sequence ends with the given [suffix], returns a new char sequence with the suffix removed. Otherwise, returns a new char sequence with the same characters.

`CharSequence.removeSuffix(suffix: CharSequence): CharSequence`
 * If this string ends with the given [suffix], returns a copy of this string with the suffix removed. Otherwise, returns this string.

`String.removeSuffix(suffix: CharSequence): String`
 * When this char sequence starts with the given [prefix] and ends with the given [suffix], returns a new char sequence having both the given [prefix] and [suffix] removed. Otherwise returns a new char sequence with the same characters.

`CharSequence.removeSurrounding(prefix: CharSequence, suffix: CharSequence): CharSequence`
 * Removes from a string both the given [prefix] and [suffix] if and only if it starts with the [prefix] and ends with the [suffix]. Otherwise returns this string unchanged.

`String.removeSurrounding(prefix: CharSequence, suffix: CharSequence): String`
 * When this char sequence starts with and ends with the given [delimiter], returns a new char sequence having this [delimiter] removed both from the start and end. Otherwise returns a new char sequence with the same characters.

`CharSequence.removeSurrounding(delimiter: CharSequence): CharSequence`
 * Removes the given [delimiter] string from both the start and the end of this string if and only if it starts with and ends with the [delimiter]. Otherwise returns this string unchanged.

`String.removeSurrounding(delimiter: CharSequence): String`
 * Replace part of string before the first occurrence of given delimiter with the [replacement] string. If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.

`String.replaceBefore(delimiter: Char, replacement: String, missingDelimiterValue: String = this): String`
 * Replace part of string before the first occurrence of given delimiter with the [replacement] string. If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.

`String.replaceAfter(delimiter: Char, replacement: String, missingDelimiterValue: String = this): String`
 * Replace part of string after the first occurrence of given delimiter with the [replacement] string. If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.

`String.replaceAfter(delimiter: String, replacement: String, missingDelimiterValue: String = this): String`
 * Replace part of string after the first occurrence of given delimiter with the [replacement] string. If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.

Replace part of string after the last occurrence of given delimiter with the [replacement] string. If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.

```

public fun String.replaceAfterLast(delimiter: String, replacement: String, missingDelimiterValue: String = this): String {
    val index = lastIndexOf(delimiter)
    return if (index == -1) missingDelimiterValue else replaceRange(index + delimiter.length, length, replacement)
}

```

Replace part of string after the last occurrence of given delimiter with the [replacement] string. If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.

```

public fun String.replaceAfterLast(delimiter: Char, replacement: String, missingDelimiterValue: String = this): String {
    val index = lastIndexOf(delimiter)
    return if (index == -1) missingDelimiterValue else replaceRange(index + 1, length, replacement)
}

```

Replace part of string before the last occurrence of given delimiter with the [replacement] string. If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.

```

public fun String.replaceBeforeLast(delimiter: Char, replacement: String, missingDelimiterValue: String = this): String {
    val index = lastIndexOf(delimiter)
    return if (index == -1) missingDelimiterValue else replaceRange(0, index, replacement)
}

```

Replace part of string before the last occurrence of given delimiter with the [replacement] string. If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.

```

public fun String.replaceBeforeLast(delimiter: String, replacement: String, missingDelimiterValue: String = this): String {
    val index = lastIndexOf(delimiter)
    return if (index == -1) missingDelimiterValue else replaceRange(0, index, replacement)
}

```

public fun String.replace(oldChar: Char, newChar: Char, ignoreCase: Boolean): String // JVM- and JS-specific

```

public fun String.replace(oldValue: String, newValue: String, ignoreCase: Boolean): String // JVM- and JS-specific

```

Returns a new string obtained by replacing each substring of this char sequence that matches the given regular expression with the given [replacement]. The [replacement] can consist of any combination of literal text and \$-substitutions. To treat the replacement string literally escape it with the [kotlin.text.Regex.Companion.escapeReplacement] method.

```

@kotlin.internal.InlineOnly
public inline fun CharSequence.replace(regex: Regex, replacement: String): String = regex.replace(this, replacement)

```

Returns a new string obtained by replacing each substring of this char sequence that matches the given regular expression with the result of the given function [transform] that takes [MatchResult] and returns a string to be used as a replacement for that match.

```

@kotlin.internal.InlineOnly
public inline fun CharSequence.replace(regex: Regex, noinline transform: (MatchResult) -> CharSequence): String = regex.replace(this, transform)

```

Replaces the first occurrence of the given regular expression [regex] in this char sequence with specified [replacement] expression.

```

@param replacement A replacement expression that can include substitutions. See [Regex.replaceFirst] for details.
@kotlin.internal.InlineOnly
public inline fun CharSequence.replaceFirst(regex: Regex, replacement: String): String = regex.replaceFirst(this, replacement)

```

Returns a copy of this string having its first character replaced with the result of the specified [transform], or the original string if it's empty.

```

@param transform function that takes the first character and returns the result of the transform applied to the character.
@sample samples.text.Strings.replaceFirstChar

```

```

@SinceKotlin("1.5")
@WasExperimental(ExperimentalStdlibApi::class)
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@JvmName("replaceFirstCharWithChar")
@kotlin.internal.InlineOnly
public inline fun String.replaceFirstChar(transform: (Char) -> Char): String {
    return if (isEmpty()) transform(this[0]) + substring(1) else this
}

```

Returns a copy of this string having its first character replaced with the result of the specified [transform], or the original string if it's empty.

```

@param transform function that takes the first character and returns the result of the transform applied to the character.
@sample samples.text.Strings.replaceFirstChar

```

```

@SinceKotlin("1.5")
@WasExperimental(ExperimentalStdlibApi::class)
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@JvmName("replaceFirstCharWithCharSequence")
@kotlin.internal.InlineOnly
public inline fun String.replaceFirstChar(transform: (Char) -> CharSequence): String {
    return if (isEmpty()) transform(this[0]).toString() + substring(1) else this
}

```

Returns true if this char sequence matches the given regular expression.

```

*\/@kotlin.internal.InlineOnly\npublic inline infix fun CharSequence.matches(regex: Regex): Boolean =
regex.matches(this)\n\n/**\n * Implementation of [regionMatches] for CharSequences.\n * Invoked when it's
already known that arguments are not Strings, so that no additional type checks are performed.\n */\ninternal fun
CharSequence.regionMatchesImpl(thisOffset: Int, other: CharSequence, otherOffset: Int, length: Int, ignoreCase:
Boolean): Boolean {\n    if ((otherOffset < 0) || (thisOffset < 0) || (thisOffset > this.length - length) || (otherOffset >
other.length - length)) {\n        return false\n    }\n    for (index in 0 until length) {\n        if (!this[thisOffset +
index].equals(other[otherOffset + index], ignoreCase))\n            return false\n    }\n    return true\n}\n\n/**\n *
Returns `true` if this char sequence starts with the specified character.\n */\npublic fun
CharSequence.startsWith(char: Char, ignoreCase: Boolean = false): Boolean =\n    this.length > 0 &&
this[0].equals(char, ignoreCase)\n\n/**\n * Returns `true` if this char sequence ends with the specified character.\n
*/\npublic fun CharSequence.endsWith(char: Char, ignoreCase: Boolean = false): Boolean =\n    this.length > 0 &&
this[lastIndex].equals(char, ignoreCase)\n\n/**\n * Returns `true` if this char sequence starts with the specified
prefix.\n */\npublic fun CharSequence.startsWith(prefix: CharSequence, ignoreCase: Boolean = false): Boolean {\n    if
(!ignoreCase && this is String && prefix is String)\n        return this.startsWith(prefix)\n    else\n        return
regionMatchesImpl(0, prefix, 0, prefix.length, ignoreCase)\n}\n\n/**\n * Returns `true` if a substring of this char
sequence starting at the specified offset [startIndex] starts with the specified prefix.\n */\npublic fun
CharSequence.startsWith(prefix: CharSequence, startIndex: Int, ignoreCase: Boolean = false): Boolean {\n    if
(!ignoreCase && this is String && prefix is String)\n        return this.startsWith(prefix, startIndex)\n    else\n
return regionMatchesImpl(startIndex, prefix, 0, prefix.length, ignoreCase)\n}\n\n/**\n * Returns `true` if this char
sequence ends with the specified suffix.\n */\npublic fun CharSequence.endsWith(suffix: CharSequence,
ignoreCase: Boolean = false): Boolean {\n    if (!ignoreCase && this is String && suffix is String)\n        return
this.endsWith(suffix)\n    else\n        return regionMatchesImpl(length - suffix.length, suffix, 0, suffix.length,
ignoreCase)\n}\n\n\n// common prefix and suffix\n\n/**\n * Returns the longest string `prefix` such that this char
sequence and [other] char sequence both start with this prefix,\n * taking care not to split surrogate pairs.\n * If this
and [other] have no common prefix, returns the empty string.\n */\n * @param ignoreCase `true` to ignore character
case when matching a character. By default `false`.\n * @sample samples.text.Strings.commonPrefixWith\n\n\n*/\npublic fun CharSequence.commonPrefixWith(other: CharSequence, ignoreCase: Boolean = false): String {\n    val
shortestLength = minOf(this.length, other.length)\n    var i = 0\n    while (i < shortestLength &&
this[i].equals(other[i], ignoreCase = ignoreCase)) {\n        i++\n    }\n    if (this.hasSurrogatePairAt(i - 1) ||
other.hasSurrogatePairAt(i - 1)) {\n        i--\n    }\n    return subSequence(0, i).toString()\n}\n\n/**\n * Returns the
longest string `suffix` such that this char sequence and [other] char sequence both end with this suffix,\n * taking
care not to split surrogate pairs.\n * If this and [other] have no common suffix, returns the empty string.\n */\n *
@param ignoreCase `true` to ignore character case when matching a character. By default `false`.\n * @sample
samples.text.Strings.commonSuffixWith\n\n\n*/\npublic fun CharSequence.commonSuffixWith(other: CharSequence,
ignoreCase: Boolean = false): String {\n    val thisLength = this.length\n    val otherLength = other.length\n    val
shortestLength = minOf(thisLength, otherLength)\n    var i = 0\n    while (i < shortestLength && this[thisLength -
i - 1].equals(other[otherLength - i - 1], ignoreCase = ignoreCase)) {\n        i++\n    }\n    if
(this.hasSurrogatePairAt(thisLength - i - 1) || other.hasSurrogatePairAt(otherLength - i - 1)) {\n        i--\n    }\n
return subSequence(thisLength - i, thisLength).toString()\n}\n\n\n// indexOfAny()\n\n/**\n * Finds the index of the
first occurrence of any of the specified [chars] in this char sequence,\n * starting from the specified [startIndex] and
optionally ignoring the case.\n */\n * @param ignoreCase `true` to ignore character case when matching a character.
By default `false`.\n * @return An index of the first occurrence of matched character from [chars] or -1 if none of
[chars] are found.\n */\n *\/\npublic fun CharSequence.indexOfAny(chars: CharArray, startIndex: Int = 0, ignoreCase:
Boolean = false): Int {\n    if (!ignoreCase && chars.size == 1 && this is String) {\n        val char = chars.single()\n
return nativeIndexOf(char, startIndex)\n    }\n    for (index in startIndex.coerceAtLeast(0)..lastIndex) {\n        val
charAtIndex = get(index)\n        if (chars.any { it.equals(charAtIndex, ignoreCase) })\n            return index\n    }\n
return -1\n}\n\n/**\n * Finds the index of the last occurrence of any of the specified [chars] in this char sequence,\n
* starting from the specified [startIndex] and optionally ignoring the case.\n */\n *\/\n * @param startIndex The index of

```

character to start searching at. The search proceeds backward toward the beginning of the string.

@param ignoreCase `true` to ignore character case when matching a character. By default `false`.

@return An index of the last occurrence of matched character from [chars] or -1 if none of [chars] are found.

```

public fun CharSequence.lastIndexOfAny(chars: CharArray, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int {
    if (!ignoreCase && chars.size == 1 && this is String) {
        val char = chars.single()
        return nativeLastIndexOf(char, startIndex)
    }
    for (index in startIndex.coerceAtMost(lastIndex) downTo 0) {
        val charAtIndex = get(index)
        if (chars.any { it.equals(charAtIndex, ignoreCase) }) return index
    }
    return -1
}

private fun CharSequence.indexOf(other: CharSequence, startIndex: Int, endIndex: Int, ignoreCase: Boolean, last: Boolean = false): Int {
    val indices = if (!last) startIndex.coerceAtLeast(0)..endIndex.coerceAtMost(length) else startIndex.coerceAtMost(lastIndex) downTo endIndex.coerceAtLeast(0)
    if (this is String && other is String) { // smart cast
        for (index in indices) {
            if (other.regionMatches(0, this, index, other.length, ignoreCase)) return index
        }
    } else {
        for (index in indices) {
            if (other.regionMatchesImpl(0, this, index, other.length, ignoreCase)) return index
        }
    }
    return -1
}

private fun CharSequence.findAnyOf(strings: Collection<String>, startIndex: Int, ignoreCase: Boolean, last: Boolean): Pair<Int, String>? {
    if (!ignoreCase && strings.size == 1) {
        val string = strings.single()
        val index = if (!last) indexOf(string, startIndex) else lastIndexOf(string, startIndex)
        return if (index < 0) null else index to string
    }
    val indices = if (!last) startIndex.coerceAtLeast(0)..length else startIndex.coerceAtMost(lastIndex) downTo 0
    if (this is String) {
        for (index in indices) {
            val matchingString = strings.firstOrNull { it.regionMatches(0, this, index, it.length, ignoreCase) }
            if (matchingString != null) return index to matchingString
        }
    } else {
        for (index in indices) {
            val matchingString = strings.firstOrNull { it.regionMatchesImpl(0, this, index, it.length, ignoreCase) }
            if (matchingString != null) return index to matchingString
        }
    }
    return null
}

```

findAnyOf Finds the first occurrence of any of the specified [strings] in this char sequence, starting from the specified [startIndex] and optionally ignoring the case.

@param ignoreCase `true` to ignore character case when matching a string. By default `false`.

@return A pair of an index of the first occurrence of matched string from [strings] and the string matched or `null` if none of [strings] are found.

To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from the beginning to the end of this string, and finds at each position the first element in [strings] that matches this string at that position.

findAnyOf Finds the last occurrence of any of the specified [strings] in this char sequence, starting from the specified [startIndex] and optionally ignoring the case.

@param startIndex The index of character to start searching at. The search proceeds backward toward the beginning of the string.

@param ignoreCase `true` to ignore character case when matching a string. By default `false`.

@return A pair of an index of the last occurrence of matched string from [strings] and the string matched or `null` if none of [strings] are found.

To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from the end toward the beginning of this string, and finds at each position the first element in [strings] that matches this string at that position.

findLastAnyOf Finds the index of the first occurrence of any of the specified [strings] in this char sequence, starting from the specified [startIndex] and optionally ignoring the case.

@param ignoreCase `true` to ignore character case when matching a string. By default `false`.

@return An index of the first occurrence of matched string from [strings] or -1 if none of [strings] are found.

To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from the beginning to the end of this string, and finds at each position the first element in [strings] that matches this string at that position.

indexOfAny Finds the index of the last occurrence of any of the specified [strings] in this char sequence, starting from the specified [startIndex] and optionally

ignoring the case.
`@param startIndex` The index of character to start searching at. The search proceeds backward toward the beginning of the string.
`@param ignoreCase`true`` to ignore character case when matching a string. By default ``false``.
`@return` An index of the last occurrence of matched string from [strings] or -1 if none of [strings] are found.
To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from the end toward the beginning of this string, and finds at each position the first element in [strings] that matches this string at that position.
`public fun`

`CharSequence.lastIndexOfAny(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int` = `findAnyOf(strings, startIndex, ignoreCase, last = true)?.first ?: -1`
Returns the index within this string of the first occurrence of the specified character, starting from the specified [startIndex].
`@param ignoreCase`true`` to ignore character case when matching a character. By default ``false``.
`@return` An index of the first occurrence of [char] or -1 if none is found.
`public fun`

`CharSequence.indexOf(char: Char, startIndex: Int = 0, ignoreCase: Boolean = false): Int` {
return if (ignoreCase || this !is String)
 indexOfAny(charArrayOf(char), startIndex, ignoreCase)
else
 nativeIndexOf(char, startIndex)
}
Returns the index within this char sequence of the first occurrence of the specified [string], starting from the specified [startIndex].
`@param ignoreCase`true`` to ignore character case when matching a string. By default ``false``.
`@return` An index of the first occurrence of [string] or -1 if none is found.
`@sample` samples.text.Strings.indexOf
`public fun` `CharSequence.indexOf(string: String, startIndex: Int = 0, ignoreCase: Boolean = false): Int` {
return if (ignoreCase || this !is String)
 indexOf(string, startIndex, length, ignoreCase)
else
 nativeIndexOf(string, startIndex)
}
Returns the index within this char sequence of the last occurrence of the specified character, starting from the specified [startIndex].
`@param startIndex` The index of character to start searching at. The search proceeds backward toward the beginning of the string.
`@param ignoreCase`true`` to ignore character case when matching a character. By default ``false``.
`@return` An index of the last occurrence of [char] or -1 if none is found.
`public fun` `CharSequence.lastIndexOf(char: Char, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int` {
return if (ignoreCase || this !is String)
 lastIndexOfAny(charArrayOf(char), startIndex, ignoreCase)
else
 nativeLastIndexOf(char, startIndex)
}
Returns the index within this char sequence of the last occurrence of the specified [string], starting from the specified [startIndex].
`@param startIndex` The index of character to start searching at. The search proceeds backward toward the beginning of the string.
`@param ignoreCase`true`` to ignore character case when matching a string. By default ``false``.
`@return` An index of the last occurrence of [string] or -1 if none is found.
`public fun` `CharSequence.lastIndexOf(string: String, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int` {
return if (ignoreCase || this !is String)
 indexOf(string, startIndex, 0, ignoreCase, last = true)
else
 nativeLastIndexOf(string, startIndex)
}
Returns ``true`` if this char sequence contains the specified [other] sequence of characters as a substring.
`@param ignoreCase`true`` to ignore character case when comparing strings. By default ``false``.
`@Suppress("INAPPLICABLE_OPERATOR_MODIFIER")`
`public operator fun`

`CharSequence.contains(other: CharSequence, ignoreCase: Boolean = false): Boolean` = `if (other is String) indexOf(other, ignoreCase = ignoreCase) >= 0 else indexOf(other, 0, length, ignoreCase) >= 0`
Returns ``true`` if this char sequence contains the specified character [char].
`@param ignoreCase`true`` to ignore character case when comparing characters. By default ``false``.
`@Suppress("INAPPLICABLE_OPERATOR_MODIFIER")`
`public operator fun` `CharSequence.contains(char: Char, ignoreCase: Boolean = false): Boolean` = `indexOf(char, ignoreCase = ignoreCase) >= 0`
Returns ``true`` if this char sequence contains at least one match of the specified regular expression [regex].
`@kotlin.internal.InlineOnly`
`public inline operator fun` `CharSequence.contains(regex: Regex): Boolean` = `regex.containsMatchIn(this)`
`private class` `DelimitedRangesSequence` {
private input: CharSequence,
private val startIndex: Int,
private val limit: Int,
private val getNextMatch: CharSequence.(current: Int) -> Pair<Int, Int>? }
Sequence<IntRange> {
override fun iterator(): Iterator<IntRange> = object : Iterator<IntRange> {
var nextState: Int = -1 // -1 for unknown, 0 for done, 1 for continue
var currentStartIndex: Int = startIndex.coerceIn(0, input.length)
var nextSearchIndex: Int =


```

currentStartIndex\n    var nextItem: IntRange? = null\n    var counter: Int = 0\n\n    private fun calcNext() {\n
        if (nextSearchIndex < 0) {\n            nextState = 0\n            nextItem = null\n        } else {\n            if
(limit > 0 && ++counter >= limit || nextSearchIndex > input.length) {\n                nextItem =
currentStartIndex..input.lastIndex\n                nextSearchIndex = -1\n            } else {\n                val match =
input.getNextMatch(nextSearchIndex)\n                if (match == null) {\n                    nextItem =
currentStartIndex..input.lastIndex\n                    nextSearchIndex = -1\n                } else {\n                    val
(index, length) = match\n                    nextItem = currentStartIndex until index\n                    currentStartIndex
= index + length\n                    nextSearchIndex = currentStartIndex + if (length == 0) 1 else 0\n                }\n
            }\n            nextState = 1\n        }\n    }\n\n    override fun next(): IntRange {\n        if (nextState ==
-1)\n            calcNext()\n        if (nextState == 0)\n            throw NoSuchElementException()\n        val
result = nextItem as IntRange\n        // Clean next to avoid keeping reference on yielded instance\n
nextItem = null\n        nextState = -1\n        return result\n    }\n\n    override fun hasNext(): Boolean {\n
        if (nextState == -1)\n            calcNext()\n        return nextState == 1\n    }\n}\n\n/**\n * Returns a
sequence of index ranges of substrings in this char sequence around occurrences of the specified [delimiters].\n *\n *
@param delimiters One or more characters to be used as delimiters.\n * @param startIndex The index to start
searching delimiters from.\n * No range having its start value less than [startIndex] is returned.\n * [startIndex] is
coerced to be non-negative and not greater than length of this string.\n * @param ignoreCase `true` to ignore
character case when matching a delimiter. By default `false`.\n * @param limit The maximum number of substrings
to return. Zero by default means no limit is set.\n */\nprivate fun CharSequence.rangesDelimitedBy(delimiters:
CharArray, startIndex: Int = 0, ignoreCase: Boolean = false, limit: Int = 0): Sequence<IntRange> {\n
    requireNonNegativeLimit(limit)\n    return DelimitedRangesSequence(this, startIndex, limit, { currentIndex ->\n
        indexOfAny(delimiters, currentIndex, ignoreCase = ignoreCase).let { if (it < 0) null else it to 1 }\n
})\n}\n\n/**\n * Returns a sequence of index ranges of substrings in this char sequence around occurrences of the
specified [delimiters].\n *\n * @param delimiters One or more strings to be used as delimiters.\n * @param
startIndex The index to start searching delimiters from.\n * No range having its start value less than [startIndex] is
returned.\n * [startIndex] is coerced to be non-negative and not greater than length of this string.\n * @param
ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.\n * @param limit The
maximum number of substrings to return. Zero by default means no limit is set.\n *\n * To avoid ambiguous results
when strings in [delimiters] have characters in common, this method proceeds from\n * the beginning to the end of
this string, and finds at each position the first element in [delimiters]\n * that matches this string at that position.\n
*/\nprivate fun CharSequence.rangesDelimitedBy(delimiters: Array<out String>, startIndex: Int = 0, ignoreCase:
Boolean = false, limit: Int = 0): Sequence<IntRange> {\n
    requireNonNegativeLimit(limit)\n    val delimitersList =
delimiters.asList()\n    return DelimitedRangesSequence(this, startIndex, limit, { currentIndex ->\n
        findAnyOf(delimitersList, currentIndex, ignoreCase = ignoreCase, last = false)?.let { it.first to it.second.length }\n
})\n}\n\ninternal fun requireNonNegativeLimit(limit: Int) =\n    require(limit >= 0) { "Limit must be non-
negative, but was $limit" }\n\n// split\n\n/**\n * Splits this char sequence to a sequence of strings around
occurrences of the specified [delimiters].\n *\n * @param delimiters One or more strings to be used as delimiters.\n
*\n * @param ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.\n * @param
limit The maximum number of substrings to return. Zero by default means no limit is set.\n *\n * To avoid
ambiguous results when strings in [delimiters] have characters in common, this method proceeds from\n * the
beginning to the end of this string, and finds at each position the first element in [delimiters]\n * that matches
this string at that position.\n */\npublic fun CharSequence.splitToSequence(vararg delimiters: String, ignoreCase:
Boolean = false, limit: Int = 0): Sequence<String> =\n    rangesDelimitedBy(delimiters, ignoreCase = ignoreCase,
limit = limit).map { substring(it) }\n\n/**\n * Splits this char sequence to a list of strings around occurrences of
the specified [delimiters].\n *\n * @param delimiters One or more strings to be used as delimiters.\n * @param
ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.\n * @param limit The
maximum number of substrings to return. Zero by default means no limit is set.\n *\n * To avoid ambiguous results
when strings in [delimiters] have characters in common, this method proceeds from\n * the beginning to the end of

```

this string, and matches at each position the first element in [delimiters]\n * that is equal to a delimiter in this instance at that position.\n *\n\npublic fun CharSequence.split(vararg delimiters: String, ignoreCase: Boolean = false, limit: Int = 0): List<String> {\n if (delimiters.size == 1) {\n val delimiter = delimiters[0]\n if (!delimiter.isEmpty()) {\n return split(delimiter, ignoreCase, limit)\n }\n }\n\n return rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit = limit).asIterable().map { substring(it) }\n}\n\n/**\n * Splits this char sequence to a sequence of strings around occurrences of the specified [delimiters].\n *\n * @param delimiters One or more characters to be used as delimiters.\n *\n * @param ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.\n *\n * @param limit The maximum number of substrings to return.\n *\n\npublic fun CharSequence.splitToSequence(vararg delimiters: Char, ignoreCase: Boolean = false, limit: Int = 0): Sequence<String> =\n rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit = limit).map { substring(it) }\n}\n\n/**\n * Splits this char sequence to a list of strings around occurrences of the specified [delimiters].\n *\n * @param delimiters One or more characters to be used as delimiters.\n *\n * @param ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.\n *\n * @param limit The maximum number of substrings to return.\n *\n\npublic fun CharSequence.split(vararg delimiters: Char, ignoreCase: Boolean = false, limit: Int = 0): List<String> {\n if (delimiters.size == 1) {\n return split(delimiters[0].toString(), ignoreCase, limit)\n }\n\n return rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit = limit).asIterable().map { substring(it) }\n}\n\n/**\n * Splits this char sequence to a list of strings around occurrences of the specified [delimiter].\n *\n * This is specialized version of split which receives single non-empty delimiter and offers better performance.\n *\n * @param delimiter String used as delimiter.\n *\n * @param ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.\n *\n * @param limit The maximum number of substrings to return.\n *\n\nprivate fun CharSequence.split(delimiter: String, ignoreCase: Boolean, limit: Int): List<String> {\n requireNonNegativeLimit(limit)\n\n var currentOffset = 0\n var nextIndex = indexOf(delimiter, currentOffset, ignoreCase)\n\n if (nextIndex == -1 || limit == 1) {\n return listOf(this.toString())\n }\n\n val isLimited = limit > 0\n val result = ArrayList<String>(if (isLimited) limit.coerceAtMost(10) else 10)\n do {\n result.add(substring(currentOffset, nextIndex))\n currentOffset = nextIndex + delimiter.length\n // Do not search for next occurrence if we're reaching limit\n if (isLimited && result.size == limit - 1) break\n\n nextIndex = indexOf(delimiter, currentOffset, ignoreCase)\n } while (nextIndex != -1)\n\n result.add(substring(currentOffset, length))\n return result\n}\n\n/**\n * Splits this char sequence to a list of strings around matches of the given regular expression.\n *\n * @param limit Non-negative value specifying the maximum number of substrings to return.\n *\n * Zero by default means no limit is set.\n *\n\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.split(regex: Regex, limit: Int = 0): List<String> =\n regex.split(this, limit)\n\n/**\n * Splits this char sequence to a sequence of strings around matches of the given regular expression.\n *\n * @param limit Non-negative value specifying the maximum number of substrings to return.\n *\n * Zero by default means no limit is set.\n *\n * @sample samples.text.Strings.splitToSequence\n *\n\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.splitToSequence(regex: Regex, limit: Int = 0): Sequence<String> =\n regex.splitToSequence(this, limit)\n\n/**\n * Splits this char sequence to a sequence of lines delimited by any of the following character sequences: CRLF, LF or CR.\n *\n * The lines returned do not include terminating line separators.\n *\n\npublic fun CharSequence.lineSequence(): Sequence<String> = splitToSequence("\\r\\n", "\\n", "\\r")\n\n/**\n * Splits this char sequence to a list of lines delimited by any of the following character sequences: CRLF, LF or CR.\n *\n * The lines returned do not include terminating line separators.\n *\n\npublic fun CharSequence.lines(): List<String> = lineSequence().toList()\n\n/**\n * Returns `true` if the contents of this char sequence are equal to the contents of the specified [other],\n * i.e. both char sequences contain the same number of the same characters in the same order.\n *\n * @sample samples.text.Strings.contentEquals\n *\n\n@SinceKotlin("1.5")\npublic expect infix fun CharSequence?.contentEquals(other: CharSequence?): Boolean\n\n/**\n * Returns `true` if the contents of this char sequence are equal to the contents of the specified [other], optionally ignoring case difference.\n *\n * @param ignoreCase `true` to ignore character case when comparing contents.\n *\n * @sample samples.text.Strings.contentEquals\n *\n\n@SinceKotlin("1.5")\npublic

```

expect fun CharSequence?.contentEquals(other: CharSequence?, ignoreCase: Boolean): Boolean\n\ninternal fun
CharSequence?.contentEqualsIgnoreCaseImpl(other: CharSequence?): Boolean {\n    if (this is String && other is
String) {\n        return this.equals(other, ignoreCase = true)\n    }\n    if (this === other) return true\n    if (this ==
null || other == null || this.length != other.length) return false\n    for (i in 0 until length) {\n        if
(!this[i].equals(other[i], ignoreCase = true)) {\n            return false\n        }\n    }\n    return true\n}\n\ninternal fun
CharSequence?.contentEqualsImpl(other: CharSequence?): Boolean {\n    if (this is String && other is String) {\n
return this == other\n    }\n    if (this === other) return true\n    if (this == null || other == null || this.length !=
other.length) return false\n    for (i in 0 until length) {\n        if (this[i] != other[i]) {\n            return false\n        }\n    }\n    return true\n}\n\n**\n * Returns `true` if the content of this string is equal to the word `true`, `false` if it is
equal to `false`,\n * and throws an exception otherwise.\n * There is also a lenient version of the function
available on nullable String, [String?.toBoolean].\n * Note that this function is case-sensitive.\n *\n * @sample
samples.text.Strings.toBooleanStrict\n *\n@SinceKotlin("1.5")\npublic fun String.toBooleanStrict(): Boolean =
when (this) {\n    "true" -> true\n    "false" -> false\n    else -> throw IllegalArgumentException("The string
doesn't represent a boolean value: $this")\n}\n\n**\n * Returns `true` if the content of this string is equal to the
word `true`, `false` if it is equal to `false`,\n * and `null` otherwise.\n *\n * There is also a lenient version of the
function available on nullable String, [String?.toBoolean].\n * Note that this function is case-sensitive.\n *\n *
@sample samples.text.Strings.toBooleanStrictOrNull\n *\n@SinceKotlin("1.5")\npublic fun
String.toBooleanStrictOrNull(): Boolean? = when (this) {\n    "true" -> true\n    "false" -> false\n    else ->
null\n},"/>\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n\n//
Auto-generated file. DO NOT EDIT!\n\npackage kotlin\n\nimport
kotlin.jvm.*\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@JvmInline\npublic value class
UByteArray\n@PublishedApi\ninternal constructor(@PublishedApi internal val storage: ByteArray) :
Collection<UByte> {\n    /** Creates a new array of the specified [size], with all elements initialized to zero. *\n
public constructor(size: Int) : this(ByteArray(size))\n    /**\n     * Returns the array element at the given [index].
This method can be called using the index operator.\n     *\n     * If the [index] is out of bounds of this array, throws
an [IndexOutOfBoundsException] except in Kotlin/JS\n     * where the behavior is unspecified.\n     *\n     public
operator fun get(index: Int): UByte = storage[index].toUByte()\n    /**\n     * Sets the element at the given [index]
to the given [value]. This method can be called using the index operator.\n     *\n     * If the [index] is out of bounds
of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n     * where the behavior is
unspecified.\n     *\n     public operator fun set(index: Int, value: UByte) {\n        storage[index] = value.toByte()\n    }\n    /** Returns the number of elements in the array. *\n     public override val size: Int get() = storage.size\n    /** Creates an iterator over the elements of the array. *\n     public override operator fun iterator():
kotlin.collections.Iterator<UByte> = Iterator(storage)\n    private class Iterator(private val array: ByteArray) :
kotlin.collections.Iterator<UByte> {\n        private var index = 0\n        override fun hasNext() = index < array.size\n            override fun next() = if (index < array.size) array[index++].toUByte() else throw
NoSuchElementException(index.toString())\n    }\n    override fun contains(element: UByte): Boolean {\n        //
TODO: Eliminate this check after KT-30016 gets fixed.\n        // Currently JS BE does not generate special bridge
method for this method.\n        @Suppress("USELESS_CAST")\n        if ((element as Any?) !is UByte) return
false\n        return storage.contains(element.toByte())\n    }\n    override fun containsAll(elements:
Collection<UByte>): Boolean {\n        return (elements as Collection<*>).all { it is UByte &&
storage.contains(it.toByte()) }\n    }\n    override fun isEmpty(): Boolean = this.storage.size == 0\n}\n\n**\n *
Creates a new array of the specified [size], where each element is calculated by calling the specified\n * [init]
function.\n *\n * The function [init] is called for each array element sequentially starting from the first one.\n * It
should return the value for an array element given its index.\n\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotl.in.internal.InlineOnly\npublic inline fun
UByteArray(size: Int, init: (Int) -> UByte): UByteArray {\n    return UByteArray(ByteArray(size) { index ->
init(index).toByte()

```

```

})\n}\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ubyteArrayOf(vararg elements: UByte): UByteArray = elements\n", "/*\n * Copyright 2010-2022 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage
kotlin\n\nimport kotlin.jvm.*\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@JvmInline\npublic
value class UIntArray\n@PublishedApi\ninternal constructor(@PublishedApi internal val storage: IntArray) :
Collection<UInt> {\n\n    /** Creates a new array of the specified [size], with all elements initialized to zero. */\n
public constructor(size: Int) : this(IntArray(size))\n\n    /**\n     * Returns the array element at the given [index].
This method can be called using the index operator.\n     *\n     * If the [index] is out of bounds of this array, throws
an [IndexOutOfBoundsException] except in Kotlin/JS\n     * where the behavior is unspecified.\n     */\n    public
operator fun get(index: Int): UInt = storage[index].toUInt()\n\n    /**\n     * Sets the element at the given [index] to
the given [value]. This method can be called using the index operator.\n     *\n     * If the [index] is out of bounds
of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n     * where the behavior is unspecified.\n
*/\n    public operator fun set(index: Int, value: UInt) {\n        storage[index] = value.toInt()\n    }\n\n    /** Returns
the number of elements in the array. */\n    public override val size: Int get() = storage.size\n\n    /** Creates an
iterator over the elements of the array. */\n    public override operator fun iterator(): kotlin.collections.Iterator<UInt>
= Iterator(storage)\n\n    private class Iterator(private val array: IntArray) : kotlin.collections.Iterator<UInt> {\n
private var index = 0\n        override fun hasNext() = index < array.size\n        override fun next() = if (index <
array.size) array[index++].toUInt() else throw NoSuchElementException(index.toString())\n    }\n\n    override fun
contains(element: UInt): Boolean {\n        // TODO: Eliminate this check after KT-30016 gets fixed.\n        //
Currently JS BE does not generate special bridge method for this method.\n
@Suppress("USELESS_CAST")\n        if ((element as Any?) !is UInt) return false\n        return
storage.contains(element.toInt())\n    }\n\n    override fun containsAll(elements: Collection<UInt>): Boolean {\n
return (elements as Collection<*>).all { it is UInt && storage.contains(it.toInt()) }\n    }\n\n    override fun
isEmpty(): Boolean = this.storage.size == 0\n}\n\n/**\n * Creates a new array of the specified [size], where each
element is calculated by calling the specified\n * [init] function.\n * The function [init] is called for each array
element sequentially starting from the first one.\n * It should return the value for an array element given its index.\n
*/\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray(size: Int, init: (Int) -> UInt): UIntArray {\n    return UIntArray(IntArray(size) { index ->
init(index).toInt()
})\n}\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
uintArrayOf(vararg elements: UInt): UIntArray = elements\n", "/*\n * Copyright 2010-2022 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage
kotlin\n\nimport kotlin.jvm.*\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@JvmInline\npublic
value class ULongArray\n@PublishedApi\ninternal constructor(@PublishedApi internal val storage: LongArray) :
Collection<ULong> {\n\n    /** Creates a new array of the specified [size], with all elements initialized to zero. */\n
public constructor(size: Int) : this(LongArray(size))\n\n    /**\n     * Returns the array element at the given [index].
This method can be called using the index operator.\n     *\n     * If the [index] is out of bounds of this array, throws
an [IndexOutOfBoundsException] except in Kotlin/JS\n     * where the behavior is unspecified.\n     */\n    public
operator fun get(index: Int): ULong = storage[index].toULong()\n\n    /**\n     * Sets the element at the given
[index] to the given [value]. This method can be called using the index operator.\n     *\n     * If the [index] is out
of bounds of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n     * where the behavior is
unspecified.\n     */\n    public operator fun set(index: Int, value: ULong) {\n        storage[index] = value.toLong()\n
    }\n\n    /** Returns the number of elements in the array. */\n    public override val size: Int get() = storage.size\n
\n    /** Creates an iterator over the elements of the array. */\n    public override operator fun iterator():
kotlin.collections.Iterator<ULong> = Iterator(storage)\n\n    private class Iterator(private val array: LongArray) :
kotlin.collections.Iterator<ULong> {\n        private var index = 0\n        override fun hasNext() = index < array.size\n
}

```

```

        override fun next() = if (index < array.size) array[index++].toULong() else throw
NoSuchElementException(index.toString())\n    }\n\n    override fun contains(element: ULong): Boolean {\n    //
TODO: Eliminate this check after KT-30016 gets fixed.\n    // Currently JS BE does not generate special bridge
method for this method.\n    @Suppress("USELESS_CAST")\n    if ((element as Any?) !is ULong) return
false\n\n    return storage.contains(element.toLong())\n    }\n\n    override fun containsAll(elements:
Collection<ULong>): Boolean {\n    return (elements as Collection<*>).all { it is ULong &&
storage.contains(it.toLong()) }\n    }\n\n    override fun isEmpty(): Boolean = this.storage.size == 0\n}\n\n/*\n *
Creates a new array of the specified [size], where each element is calculated by calling the specified\n * [init]
function.\n * The function [init] is called for each array element sequentially starting from the first one.\n * It
should return the value for an array element given its index.\n
*/\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray(size: Int, init: (Int) -> ULong): ULongArray {\n    return ULongArray(LongArray(size) { index ->
init(index).toLong()
})\n}\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ulongArrayOf(vararg elements: ULong): ULongArray = elements\n"/*\n * Copyright 2010-2022 JetBrains s.r.o.
and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage
kotlin\n\nimport kotlin.jvm.*\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@JvmInline\npublic
value class UShortArray\n@PublishedApi\ninternal constructor(@PublishedApi internal val storage: ShortArray) :
Collection<UShort> {\n\n    /** Creates a new array of the specified [size], with all elements initialized to zero. */\n
public constructor(size: Int) : this(ShortArray(size))\n\n    /**\n     * Returns the array element at the given [index].
This method can be called using the index operator.\n     * If the [index] is out of bounds of this array, throws
an [IndexOutOfBoundsException] except in Kotlin/JS\n     * where the behavior is unspecified.\n     */\n    public
operator fun get(index: Int): UShort = storage[index].toUShort()\n\n    /**\n     * Sets the element at the given
[index] to the given [value]. This method can be called using the index operator.\n     * If the [index] is out of
bounds of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n     * where the behavior is
unspecified.\n     */\n    public operator fun set(index: Int, value: UShort) {\n        storage[index] = value.toShort()
}\n\n    /** Returns the number of elements in the array. */\n    public override val size: Int get() = storage.size\n\n
/** Creates an iterator over the elements of the array. */\n    public override operator fun iterator():
kotlin.collections.Iterator<UShort> = Iterator(storage)\n\n    private class Iterator(private val array: ShortArray) :
kotlin.collections.Iterator<UShort> {\n        private var index = 0\n        override fun hasNext() = index <
array.size\n        override fun next() = if (index < array.size) array[index++].toUShort() else throw
NoSuchElementException(index.toString())\n    }\n\n    override fun contains(element: UShort): Boolean {\n    //
TODO: Eliminate this check after KT-30016 gets fixed.\n    // Currently JS BE does not generate special bridge
method for this method.\n    @Suppress("USELESS_CAST")\n    if ((element as Any?) !is UShort) return
false\n\n    return storage.contains(element.toShort())\n    }\n\n    override fun containsAll(elements:
Collection<UShort>): Boolean {\n    return (elements as Collection<*>).all { it is UShort &&
storage.contains(it.toShort()) }\n    }\n\n    override fun isEmpty(): Boolean = this.storage.size == 0\n}\n\n/*\n *
Creates a new array of the specified [size], where each element is calculated by calling the specified\n * [init]
function.\n * The function [init] is called for each array element sequentially starting from the first one.\n * It
should return the value for an array element given its index.\n
*/\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray(size: Int, init: (Int) -> UShort): UShortArray {\n    return UShortArray(ShortArray(size) { index ->
init(index).toShort()
})\n}\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ushortArrayOf(vararg elements: UShort): UShortArray = elements\n"/*\n * Copyright 2010-2022 JetBrains s.r.o.
and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n

```


If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun  
UShortArray.component3(): UShort {\n    return get(2)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n *
```

If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun  
UIntArray.component4(): UInt {\n    return get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n *
```

If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic  
inline operator fun  
ULongArray.component4(): ULong {\n    return get(3)\n}\n\n/**\n * Returns 4th *element* from  
the array.\n * \n *
```

If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun  
UByteArray.component4(): UByte {\n    return get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n *
```

If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun  
UShortArray.component4(): UShort {\n    return get(3)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n *
```

If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun  
UIntArray.component5(): UInt {\n    return get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n *
```

If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic  
inline operator fun  
ULongArray.component5(): ULong {\n    return get(4)\n}\n\n/**\n * Returns 5th *element* from  
the array.\n * \n *
```

If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun  
UByteArray.component5(): UByte {\n    return get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n *
```

If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun  
UShortArray.component5(): UShort {\n    return get(4)\n}\n\n/**\n * Returns an element at the given [index] or  
throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample  
samples.collections.Collections.Elements.elementAt\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun UIntArray.elementAt(index: Int):  
UInt\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is  
out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun ULongArray.elementAt(index: Int):  
ULong\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index]  
is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun UByteArray.elementAt(index: Int):  
UByte\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index]  
is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun UShortArray.elementAt(index: Int):  
UShort\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the  
[index] is out of bounds of this array.\n * \n * @sample  
samples.collections.Collections.Elements.elementAt\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun UShortArray.elementAt(index: Int):  
UShort\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the  
[index] is out of bounds of this array.\n * \n * @sample  
samples.collections.Collections.Elements.elementAt\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun UShortArray.elementAt(index: Int):  
UShort\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the  
[index] is out of bounds of this array.\n * \n * @sample  
samples.collections.Collections.Elements.elementAt\n
```

```

samples.collections.Collections.Elements.elementAtOrElse\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UInt): UInt {\n  return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.elementAtOrElse(index: Int, defaultValue: (Int) -> ULong): ULong {\n  return if (index >= 0 &&
index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the
result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UByte): UByte {\n  return if (index >= 0 && index
<= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result
of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UShort): UShort {\n  return if (index >= 0 &&
index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or
`null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.elementAtOrNull(index: Int): UInt? {\n  return this.getOrNull(index)\n}\n\n/**\n * Returns an element
at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.elementAtOrNull(index: Int): ULong? {\n  return this.getOrNull(index)\n}\n\n/**\n * Returns an
element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.elementAtOrNull(index: Int): UByte? {\n  return this.getOrNull(index)\n}\n\n/**\n * Returns an
element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.elementAtOrNull(index: Int): UShort? {\n  return this.getOrNull(index)\n}\n\n/**\n * Returns the
first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.find(predicate: (UInt) -> Boolean): UInt? {\n  return firstOrNull(predicate)\n}\n\n/**\n * Returns the
first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.find(predicate: (ULong) -> Boolean): ULong? {\n  return firstOrNull(predicate)\n}\n\n/**\n *
Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.find(predicate: (UByte) -> Boolean): UByte? {\n  return firstOrNull(predicate)\n}\n\n/**\n * Returns
the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample

```



```

samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.find(predicate: (UShort) -> Boolean): UShort? {\n    return firstOrNull(predicate)\n}\n\n/**\n *
Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.findLast(predicate: (UInt) -> Boolean): UInt? {\n    return lastOrNull(predicate)\n}\n\n/**\n * Returns the
last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.findLast(predicate: (ULong) -> Boolean): ULong? {\n    return lastOrNull(predicate)\n}\n\n/**\n *
Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.findLast(predicate: (UByte) -> Boolean): UByte? {\n    return lastOrNull(predicate)\n}\n\n/**\n *
Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.findLast(predicate: (UShort) -> Boolean): UShort? {\n    return lastOrNull(predicate)\n}\n\n/**\n *
Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.first(): UInt {\n    return storage.first().toUInt()\n}\n\n/**\n * Returns the first element.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.first(): ULong {\n    return storage.first().toULong()\n}\n\n/**\n * Returns the first element.\n * \n *
@throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.first(): UByte {\n    return storage.first().toUByte()\n}\n\n/**\n * Returns the first element.\n * \n *
@throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.first(): UShort {\n    return storage.first().toUShort()\n}\n\n/**\n * Returns the first element matching
the given [predicate].\n * \n * @throws [NoSuchElementException] if no such element is found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.first(predicate: (UInt) -> Boolean): UInt {\n    for (element in this) if (predicate(element)) return
element\n    throw NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n *
Returns the first element matching the given [predicate].\n * \n * @throws [NoSuchElementException] if no such
element is found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.first(predicate: (ULong) -> Boolean): ULong {\n    for (element in this) if (predicate(element)) return
element\n    throw NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n *
Returns the first element matching the given [predicate].\n * \n * @throws [NoSuchElementException] if no such
element is found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.first(predicate: (UByte) -> Boolean): UByte {\n    for (element in this) if (predicate(element)) return
element\n    throw NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n *
Returns the first element matching the given [predicate].\n * \n * @throws [NoSuchElementException] if no such
element is found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun

```

```

UShortArray.first(predicate: (UShort) -> Boolean): UShort {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}
Returns the first element, or `null` if the array is empty.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public fun UIntArray.firstOrNull(): UInt? {
    return if (isEmpty()) null else this[0]
}
Returns the first element, or `null` if the array is empty.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public fun ULongArray.firstOrNull(): ULong? {
    return if (isEmpty()) null else this[0]
}
Returns the first element, or `null` if the array is empty.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public fun UByteArray.firstOrNull(): UByte? {
    return if (isEmpty()) null else this[0]
}
Returns the first element, or `null` if the array is empty.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public fun UShortArray.firstOrNull(predicate: (UShort) -> Boolean): UShort? {
    return if (isEmpty()) null else this[0]
}
Returns the first element matching the given [predicate], or `null` if element was not found.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun UIntArray.firstOrNull(predicate: (UInt) -> Boolean): UInt? {
    for (element in this) if (predicate(element)) return element
    return null
}
Returns the first element matching the given [predicate], or `null` if element was not found.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun ULongArray.firstOrNull(predicate: (ULong) -> Boolean): ULong? {
    for (element in this) if (predicate(element)) return element
    return null
}
Returns the first element matching the given [predicate], or `null` if element was not found.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun UByteArray.firstOrNull(predicate: (UByte) -> Boolean): UByte? {
    for (element in this) if (predicate(element)) return element
    return null
}
Returns the first element matching the given [predicate], or `null` if element was not found.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun UShortArray.firstOrNull(predicate: (UShort) -> Boolean): UShort? {
    for (element in this) if (predicate(element)) return element
    return null
}
Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun UIntArray.getOrNull(index: Int, defaultValue: (Int) -> UInt): UInt {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun ULongArray.getOrNull(index: Int, defaultValue: (Int) -> ULong): ULong {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun UByteArray.getOrNull(index: Int, defaultValue: (Int) -> UByte): UByte {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun UShortArray.getOrNull(index: Int, defaultValue: (Int) -> UShort): UShort {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public fun UIntArray.getOrNull(index: Int): UInt? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}
Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public fun ULongArray.getOrNull(index: Int): ULong? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}
Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.

```

```

ULong? {\n    return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at
the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.getOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.getOrNull(index: Int): UByte?
{\n    return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the
given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.getOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.getOrNull(index: Int):
UShort? {\n    return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns first index of
[element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.indexOf(element: UInt): Int {\n    return storage.indexOf(element.toInt())\n}\n\n/**\n * Returns first
index of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.indexOf(element: ULong): Int {\n    return storage.indexOf(element.toLong())\n}\n\n/**\n * Returns
first index of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.indexOf(element: UByte): Int {\n    return storage.indexOf(element.toByte())\n}\n\n/**\n * Returns
first index of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.indexOf(element: UShort): Int {\n    return storage.indexOf(element.toShort())\n}\n\n/**\n * Returns
index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.indexOfFirst(predicate: (UInt) -> Boolean): Int {\n    return storage.indexOfFirst { predicate(it.toUInt())
}\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain
such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.indexOfFirst(predicate: (ULong) -> Boolean): Int {\n    return storage.indexOfFirst {
predicate(it.toULong()) }\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the
array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.indexOfFirst(predicate: (UByte) -> Boolean): Int {\n    return storage.indexOfFirst {
predicate(it.toUByte()) }\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the
array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.indexOfFirst(predicate: (UShort) -> Boolean): Int {\n    return storage.indexOfFirst {
predicate(it.toUShort()) }\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the
array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.indexOfLast(predicate: (UInt) -> Boolean): Int {\n    return storage.indexOfLast { predicate(it.toUInt())
}\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the array does not contain
such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.indexOfLast(predicate: (ULong) -> Boolean): Int {\n    return storage.indexOfLast {
predicate(it.toULong()) }\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the
array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.indexOfLast(predicate: (UByte) -> Boolean): Int {\n    return storage.indexOfLast {
predicate(it.toUByte()) }\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the
array does not contain such element.\n

```

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.indexOfLast(predicate: (UShort) -> Boolean): Int {\n    return storage.indexOfLast {
predicate(it.toUShort()) }\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the
array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.last(): UInt {\n    return storage.last().toUInt()\n}\n\n/**\n * Returns the last element.\n * \n * @throws
NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.last(): ULong {\n    return storage.last().toULong()\n}\n\n/**\n * Returns the last element.\n * \n *
@throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.last(): UByte {\n    return storage.last().toUByte()\n}\n\n/**\n * Returns the last element.\n * \n *
@throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.last(): UShort {\n    return storage.last().toUShort()\n}\n\n/**\n * Returns the last element matching
the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.last(predicate: (UInt) -> Boolean): UInt {\n    for (index in this.indices.reversed()) {\n        val element =
this[index]\n        if (predicate(element)) return element\n    }\n    throw NoSuchElementException("Array contains
no element matching the predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n *
@throws NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.last(predicate: (ULong) -> Boolean): ULong {\n    for (index in this.indices.reversed()) {\n        val
element = this[index]\n        if (predicate(element)) return element\n    }\n    throw
NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the last
element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n
*\n * @sample samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.last(predicate: (UByte) -> Boolean): UByte {\n    for (index in this.indices.reversed()) {\n        val
element = this[index]\n        if (predicate(element)) return element\n    }\n    throw
NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the last
element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n
*\n * @sample samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.last(predicate: (UShort) -> Boolean): UShort {\n    for (index in this.indices.reversed()) {\n        val
element = this[index]\n        if (predicate(element)) return element\n    }\n    throw
NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns last index
of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.lastIndexOf(element: UInt): Int {\n    return storage.lastIndexOf(element.toInt())\n}\n\n/**\n * Returns
last index of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.lastIndexOf(element: ULong): Int {\n    return storage.lastIndexOf(element.toLong())\n}\n\n/**\n *
Returns last index of [element], or -1 if the array does not contain element.\n

```

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.lastIndexOf(element: UByte): Int {\n    return storage.lastIndexOf(element.toByte())\n}\n\n/**\n *
Returns last index of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.lastIndexOf(element: UShort): Int {\n    return storage.lastIndexOf(element.toShort())\n}\n\n/**\n *
Returns the last element, or `null` if the array is empty.\n
*\n * @sample
samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun UIntArray.lastOrNull(): UInt? {\n    return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n *
Returns the last
element, or `null` if the array is empty.\n
*\n * @sample samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.lastOrNull(): ULong? {\n
return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n *
Returns the last element, or `null` if the array is empty.\n
*\n * @sample samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.lastOrNull(): UByte? {\n
return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n *
Returns the last element, or `null` if the array is empty.\n
*\n * @sample samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.lastOrNull(): UShort? {\n
return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n *
Returns the last element matching the given [predicate], or
`null` if no such element was found.\n
*\n * @sample samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.lastOrNull(predicate: (UInt) -> Boolean): UInt? {\n    for (index in this.indices.reversed()) {\n        val
element = this[index]\n        if (predicate(element)) return element\n    }\n    return null\n}\n\n/**\n *
Returns the last
element matching the given [predicate], or `null` if no such element was found.\n
*\n * @sample
samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.lastOrNull(predicate: (ULong) -> Boolean): ULong? {\n    for (index in this.indices.reversed()) {\n
val element = this[index]\n        if (predicate(element)) return element\n    }\n    return null\n}\n\n/**\n *
Returns the
last element matching the given [predicate], or `null` if no such element was found.\n
*\n * @sample
samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.lastOrNull(predicate: (UByte) -> Boolean): UByte? {\n    for (index in this.indices.reversed()) {\n
val element = this[index]\n        if (predicate(element)) return element\n    }\n    return null\n}\n\n/**\n *
Returns the
last element matching the given [predicate], or `null` if no such element was found.\n
*\n * @sample
samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.lastOrNull(predicate: (UShort) -> Boolean): UShort? {\n    for (index in this.indices.reversed()) {\n
val element = this[index]\n        if (predicate(element)) return element\n    }\n    return null\n}\n\n/**\n *
Returns a
random element from this array.\n
*\n * @throws NoSuchElementException if this array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.random(): UInt {\n    return random(Random)\n}\n\n/**\n *
Returns a random element from this array.\n
*\n * @throws NoSuchElementException if this array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.random(): ULong {\n    return random(Random)\n}\n\n/**\n *
Returns a random element from this
array.\n
*\n * @throws NoSuchElementException if this array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.random(): UByte {\n    return random(Random)\n}\n\n/**\n *
Returns a random element from this
array.\n
*\n * @throws NoSuchElementException if this array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.random(): UShort {\n    return random(Random)\n}\n\n/**\n *
Returns a random element from this

```

```

array using the specified source of randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.random(random: Random): UInt
{\n if (isEmpty())\n throw NoSuchElementException("Array is empty.")\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.random(random: Random):
ULong {\n if (isEmpty())\n throw NoSuchElementException("Array is empty.")\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.random(random: Random):
UByte {\n if (isEmpty())\n throw NoSuchElementException("Array is empty.")\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.random(random: Random):
UShort {\n if (isEmpty())\n throw NoSuchElementException("Array is empty.")\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UIntArray.randomOrNull(): UInt? {\n return
randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun ULongArray.randomOrNull(): ULong? {\n return
randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UByteArray.randomOrNull(): UByte? {\n return
randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UShortArray.randomOrNull(): UShort? {\n return
randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\np
ublic fun UIntArray.randomOrNull(random: Random): UInt? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\np
ublic fun ULongArray.randomOrNull(random: Random): ULong? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\np
ublic fun UByteArray.randomOrNull(random: Random): UByte? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\np
ublic fun UShortArray.randomOrNull(random: Random): UShort? {\n if (isEmpty())\n return null\n return
get(random.nextInt(size))\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or
has more than one element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.single(): UInt {\n return storage.single().toInt()\n}\n\n/**\n * Returns the single element, or throws an
exception if the array is empty or has more than one element.\n

```

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.single(): ULong {\n    return storage.single().toULong()\n}\n\n/**\n * Returns the single element, or
throws an exception if the array is empty or has more than one element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.single(): UByte {\n    return storage.single().toUByte()\n}\n\n/**\n * Returns the single element, or
throws an exception if the array is empty or has more than one element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.single(): UShort {\n    return storage.single().toUShort()\n}\n\n/**\n * Returns the single element
matching the given [predicate], or throws exception if there is no or more than one matching element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.single(predicate: (UInt) -> Boolean): UInt {\n    var single: UInt? = null\n    var found = false\n    for
(element in this) {\n        if (predicate(element)) {\n            if (found) throw IllegalArgumentException("Array
contains more than one matching element.")\n            single = element\n            found = true\n        }\n    }
if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n    @Suppress("UNCHECKED_CAST")\n    return single as UInt\n}\n\n/**\n * Returns the single element matching
the given [predicate], or throws exception if there is no or more than one matching element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.single(predicate: (ULong) -> Boolean): ULong {\n    var single: ULong? = null\n    var found = false\n
for (element in this) {\n        if (predicate(element)) {\n            if (found) throw IllegalArgumentException("Array
contains more than one matching element.")\n            single = element\n            found = true\n        }\n    }
if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n    @Suppress("UNCHECKED_CAST")\n    return single as ULong\n}\n\n/**\n * Returns the single element
matching the given [predicate], or throws exception if there is no or more than one matching element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.single(predicate: (UByte) -> Boolean): UByte {\n    var single: UByte? = null\n    var found = false\n
for (element in this) {\n        if (predicate(element)) {\n            if (found) throw IllegalArgumentException("Array
contains more than one matching element.")\n            single = element\n            found = true\n        }\n    }
if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n    @Suppress("UNCHECKED_CAST")\n    return single as UByte\n}\n\n/**\n * Returns the single element
matching the given [predicate], or throws exception if there is no or more than one matching element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.single(predicate: (UShort) -> Boolean): UShort {\n    var single: UShort? = null\n    var found = false\n
for (element in this) {\n        if (predicate(element)) {\n            if (found) throw IllegalArgumentException("Array
contains more than one matching element.")\n            single = element\n            found = true\n        }\n    }
if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n    @Suppress("UNCHECKED_CAST")\n    return single as UShort\n}\n\n/**\n * Returns single element, or `null` if
the array is empty or has more than one element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.singleOrNull(): UInt? {\n    return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more
than one element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
ULongArray.singleOrNull(): ULong? {\n    return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single
element, or `null` if the array is empty or has more than one element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UByteArray.singleOrNull(): UByte? {\n    return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more
than one element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UShortArray.singleOrNull(): UShort? {\n    return if (size == 1) this[0] else null\n}\n\n/**\n * Returns the single
element matching the given [predicate], or `null` if element was not found or more than one element was found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun

```

```

UIntArray.singleOrNull(predicate: (UInt) -> Boolean): UInt? {\n  var single: UInt? = null\n  var found = false\n  for (element in this) {\n    if (predicate(element)) {\n      if (found) return null\n      single = element\n      found = true\n    }\n  }\n  if (!found) return null\n  return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nULongArray.singleOrNull(predicate: (ULong) -> Boolean): ULong? {\n  var single: ULong? = null\n  var found = false\n  for (element in this) {\n    if (predicate(element)) {\n      if (found) return null\n      single = element\n      found = true\n    }\n  }\n  if (!found) return null\n  return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUByteArray.singleOrNull(predicate: (UByte) -> Boolean): UByte? {\n  var single: UByte? = null\n  var found = false\n  for (element in this) {\n    if (predicate(element)) {\n      if (found) return null\n      single = element\n      found = true\n    }\n  }\n  if (!found) return null\n  return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUShortArray.singleOrNull(predicate: (UShort) -> Boolean): UShort? {\n  var single: UShort? = null\n  var found = false\n  for (element in this) {\n    if (predicate(element)) {\n      if (found) return null\n      single = element\n      found = true\n    }\n  }\n  if (!found) return null\n  return single\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n */\n\n@sample\nsamples.collections.Collections.Transformations.drop\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.drop(n: Int): List<UInt> {\n  require(n >= 0) { "Requested element count $n is less than zero." }\n  return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample\nsamples.collections.Collections.Transformations.drop\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.drop(n: Int): List<ULong> {\n  require(n >= 0) { "Requested element count $n is less than zero." }\n  return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample\nsamples.collections.Collections.Transformations.drop\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.drop(n: Int): List<UByte> {\n  require(n >= 0) { "Requested element count $n is less than zero." }\n  return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample\nsamples.collections.Collections.Transformations.drop\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.drop(n: Int): List<UShort> {\n  require(n >= 0) { "Requested element count $n is less than zero." }\n  return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample\nsamples.collections.Collections.Transformations.drop\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.dropLast(n: Int): List<UInt> {\n  require(n >= 0) { "Requested element count $n is less than zero." }\n  return take((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample\nsamples.collections.Collections.Transformations.drop\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.dropLast(n: Int): List<ULong> {\n  require(n >= 0) { "Requested element count $n is less than zero." }\n  return take((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws

```



```

IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.dropLast(n: Int): List<UByte>
{\n  require(n >= 0) { "Requested element count $n is less than zero." }\n  return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.dropLast(n: Int):
List<UShort> {\n  require(n >= 0) { "Requested element count $n is less than zero." }\n  return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given
[predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.dropLastWhile(predicate: (UInt) -> Boolean): List<UInt> {\n  for (index in lastIndex downTo 0) {\n
if (!predicate(this[index])) {\n    return take(index + 1)\n  }\n  }\n  return emptyList()\n}\n\n/**\n *
Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.dropLastWhile(predicate: (ULong) -> Boolean): List<ULong> {\n  for (index in lastIndex downTo 0)
{\n  if (!predicate(this[index])) {\n    return take(index + 1)\n  }\n  }\n  return emptyList()\n}\n\n/**\n *
Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.dropLastWhile(predicate: (UByte) -> Boolean): List<UByte> {\n  for (index in lastIndex downTo 0)
{\n  if (!predicate(this[index])) {\n    return take(index + 1)\n  }\n  }\n  return emptyList()\n}\n\n/**\n *
Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.dropLastWhile(predicate: (UShort) -> Boolean): List<UShort> {\n  for (index in lastIndex downTo
0) {\n  if (!predicate(this[index])) {\n    return take(index + 1)\n  }\n  }\n  return
emptyList()\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given
[predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.dropWhile(predicate: (UInt) -> Boolean): List<UInt> {\n  var yielding = false\n  val list =
ArrayList<UInt>()\n  for (item in this)\n    if (yielding)\n      list.add(item)\n    else if (!predicate(item)) {\n
list.add(item)\n      yielding = true\n    }\n  return list\n}\n\n/**\n * Returns a list containing all
elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.dropWhile(predicate: (ULong) -> Boolean): List<ULong> {\n  var yielding = false\n  val list =
ArrayList<ULong>()\n  for (item in this)\n    if (yielding)\n      list.add(item)\n    else if (!predicate(item))
{\n  list.add(item)\n      yielding = true\n    }\n  return list\n}\n\n/**\n * Returns a list containing all
elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.dropWhile(predicate: (UByte) -> Boolean): List<UByte> {\n  var yielding = false\n  val list =
ArrayList<UByte>()\n  for (item in this)\n    if (yielding)\n      list.add(item)\n    else if (!predicate(item))
{\n  list.add(item)\n      yielding = true\n    }\n  return list\n}\n\n/**\n * Returns a list containing all
elements except first elements that satisfy the given [predicate].\n * \n * @sample

```

```

samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.dropWhile(predicate: (UShort) -> Boolean): List<UShort> {\n  var yielding = false\n  val list =
ArrayList<UShort>()\n  for (item in this)\n    if (yielding)\n      list.add(item)\n    else if (!predicate(item))\n      list.add(item)\n      yielding = true\n  }\n  return list\n}\n\n/**\n * Returns a list containing only
elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.filter(predicate: (UInt) -> Boolean): List<UInt> {\n  return filterTo(ArrayList<UInt>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.filter(predicate: (ULong) -> Boolean): List<ULong> {\n  return filterTo(ArrayList<ULong>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.filter(predicate: (UByte) -> Boolean): List<UByte> {\n  return filterTo(ArrayList<UByte>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.filter(predicate: (UShort) -> Boolean): List<UShort> {\n  return filterTo(ArrayList<UShort>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param
[predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate
evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.filterIndexed(predicate: (index: Int, UInt) -> Boolean): List<UInt> {\n  return
filterIndexedTo(ArrayList<UInt>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the
given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and
returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.filterIndexed(predicate: (index: Int, ULong) -> Boolean): List<ULong> {\n  return
filterIndexedTo(ArrayList<ULong>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the
given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and
returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.filterIndexed(predicate: (index: Int, UByte) -> Boolean): List<UByte> {\n  return
filterIndexedTo(ArrayList<UByte>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the
given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and
returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.filterIndexed(predicate: (index: Int, UShort) -> Boolean): List<UShort> {\n  return
filterIndexedTo(ArrayList<UShort>(), predicate)\n}\n\n/**\n * Appends all elements matching the given [predicate]
to the given [destination].\n * @param [predicate] function that takes the index of an element and the element
itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexedTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :

```



```

all elements not matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UByte>> UByteArray.filterNotTo(destination: C, predicate: (UByte) -> Boolean): C {\n for
(element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all
elements not matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UShort>> UShortArray.filterNotTo(destination: C, predicate: (UShort) -> Boolean): C {\n
for (element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends
all elements matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UInt>> UIntArray.filterTo(destination: C, predicate: (UInt) -> Boolean): C {\n for (element
in this) if (predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements
matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in ULong>> ULongArray.filterTo(destination: C, predicate: (ULong) -> Boolean): C {\n for
(element in this) if (predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all
elements matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UByte>> UByteArray.filterTo(destination: C, predicate: (UByte) -> Boolean): C {\n for
(element in this) if (predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all
elements matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UShort>> UShortArray.filterTo(destination: C, predicate: (UShort) -> Boolean): C {\n for
(element in this) if (predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Returns a list
containing elements at indices in the specified [indices] range.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.slice(indices: IntRange):
List<UInt> {\n if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive +
1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.slice(indices: IntRange):
List<ULong> {\n if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive
+ 1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.slice(indices: IntRange):
List<UByte> {\n if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive
+ 1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.slice(indices: IntRange):
List<UShort> {\n if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive
+ 1).asList()\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.slice(indices: Iterable<Int>):
List<UInt> {\n val size = indices.collectionSizeOrDefault(10)\n if (size == 0) return emptyList()\n val list =
ArrayList<UInt>(size)\n for (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n *
Returns a list containing elements at specified [indices].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.slice(indices: Iterable<Int>):

```

```

List<ULong> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<ULong>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

Returns a list containing elements at specified [indices].

*SinceKotlin("1.3")ExperimentalUnsignedTypes
public fun UByteArray.slice(indices: Iterable<Int>):
List<UByte> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<UByte>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

Returns a list containing elements at specified [indices].

*SinceKotlin("1.3")ExperimentalUnsignedTypes
public fun UShortArray.slice(indices: Iterable<Int>):
List<UShort> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<UShort>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

Returns an array containing elements of this array at specified [indices].

*SinceKotlin("1.3")ExperimentalUnsignedTypes
public fun UIntArray.sliceArray(indices:
Collection<Int>): UIntArray {
    return UIntArray(storage.sliceArray(indices))
}

Returns an array containing elements of this array at specified [indices].

*SinceKotlin("1.3")ExperimentalUnsignedTypes
public fun ULongArray.sliceArray(indices:
Collection<Int>): ULongArray {
    return ULongArray(storage.sliceArray(indices))
}

Returns an array containing elements of this array at specified [indices].

*SinceKotlin("1.3")ExperimentalUnsignedTypes
public fun UByteArray.sliceArray(indices:
Collection<Int>): UByteArray {
    return UByteArray(storage.sliceArray(indices))
}

Returns an array containing elements of this array at specified [indices].

*SinceKotlin("1.3")ExperimentalUnsignedTypes
public fun UShortArray.sliceArray(indices:
Collection<Int>): UShortArray {
    return UShortArray(storage.sliceArray(indices))
}

Returns an array containing elements at indices in the specified [indices] range.

*SinceKotlin("1.3")ExperimentalUnsignedTypes
public fun UIntArray.sliceArray(indices: IntRange):
UIntArray {
    return UIntArray(storage.sliceArray(indices))
}

Returns an array containing elements at indices in the specified [indices] range.

*SinceKotlin("1.3")ExperimentalUnsignedTypes
public fun ULongArray.sliceArray(indices: IntRange): ULongArray {
    return ULongArray(storage.sliceArray(indices))
}

Returns an array containing elements at indices in the specified [indices] range.

*SinceKotlin("1.3")ExperimentalUnsignedTypes
public fun UByteArray.sliceArray(indices: IntRange): UByteArray {
    return UByteArray(storage.sliceArray(indices))
}

Returns an array containing elements at indices in the specified [indices] range.

*SinceKotlin("1.3")ExperimentalUnsignedTypes
public fun UShortArray.sliceArray(indices: IntRange): UShortArray {
    return UShortArray(storage.sliceArray(indices))
}

Returns a list containing first [n] elements.

@throws IllegalArgumentException if [n] is negative.
@sample
samples.collections.Collections.Transformations.take

*SinceKotlin("1.3")ExperimentalUnsignedTypes
public fun UIntArray.take(n: Int): List<UInt> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    if (n >= size) return toList()
    if (n == 1) return listOf(this[0])
    var count = 0
    val list = ArrayList<UInt>(n)
    for (item in this) {
        list.add(item)
        if (++count == n) break
    }
    return list
}

Returns a list containing first [n] elements.

@throws IllegalArgumentException if [n] is negative.
@sample
samples.collections.Collections.Transformations.take

*SinceKotlin("1.3")ExperimentalUnsignedTypes
public fun ULongArray.take(n: Int): List<ULong> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    if (n >= size) return toList()
    if (n == 1) return listOf(this[0])
    var count = 0
    val list = ArrayList<ULong>(n)
    for (item in this) {
        list.add(item)
        if (++count == n) break
    }
    return list
}

Returns a list containing first [n] elements.

@throws IllegalArgumentException if [n] is negative.
@sample
samples.collections.Collections.Transformations.take

```

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.take(n: Int): List<UByte> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n if (n >=
size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n val list = ArrayList<UByte>(n)\n for
(item in this) {\n list.add(item)\n if (++count == n)\n break\n }\n return list\n}\n\n/**\n * Returns
a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.take(n: Int): List<UShort> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n if (n >=
size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n val list = ArrayList<UShort>(n)\n for
(item in this) {\n list.add(item)\n if (++count == n)\n break\n }\n return list\n}\n\n/**\n * Returns
a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.takeLast(n: Int): List<UInt> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n val size =
size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list = ArrayList<UInt>(n)\n
for (index in size - n until size)\n list.add(this[index])\n return list\n}\n\n/**\n * Returns a list containing last
[n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.takeLast(n: Int): List<ULong>\n
{\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n val
size = size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list =
ArrayList<ULong>(n)\n for (index in size - n until size)\n list.add(this[index])\n return list\n}\n\n/**\n *
Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n *
@sample samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.takeLast(n: Int): List<UByte>\n
{\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n val
size = size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list =
ArrayList<UByte>(n)\n for (index in size - n until size)\n list.add(this[index])\n return list\n}\n\n/**\n *
Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n *
@sample samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.takeLast(n: Int): List<UShort>\n
{\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n val
size = size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list =
ArrayList<UShort>(n)\n for (index in size - n until size)\n list.add(this[index])\n return list\n}\n\n/**\n *
Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.takeLastWhile(predicate: (UInt) -> Boolean): List<UInt> {\n for (index in lastIndex downTo 0) {\n
if (!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n * Returns a
list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.takeLastWhile(predicate: (ULong) -> Boolean): List<ULong> {\n for (index in lastIndex downTo 0)
{\n if (!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n *
Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.takeLastWhile(predicate: (UByte) -> Boolean): List<UByte> {\n for (index in lastIndex downTo 0)

```

```

{\n    if (!predicate(this[index])) {\n        return drop(index + 1)\n    }\n } \n return toList()\n}\n\n/**\n *
Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.takeLastWhile(predicate: (UShort) -> Boolean): List<UShort> {\n    for (index in lastIndex downTo 0)
{\n        if (!predicate(this[index])) {\n            return drop(index + 1)\n        }\n    }\n    return toList()\n}\n\n/**\n *
Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.takeWhile(predicate: (UInt) -> Boolean): List<UInt> {\n    val list = ArrayList<UInt>()\n    for (item in
this) {\n        if (!predicate(item))\n            break\n        list.add(item)\n    }\n    return list\n}\n\n/**\n *
Returns a list
containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.takeWhile(predicate: (ULong) -> Boolean): List<ULong> {\n    val list = ArrayList<ULong>()\n    for
(item in this) {\n        if (!predicate(item))\n            break\n        list.add(item)\n    }\n    return list\n}\n\n/**\n *
Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.takeWhile(predicate: (UByte) -> Boolean): List<UByte> {\n    val list = ArrayList<UByte>()\n    for
(item in this) {\n        if (!predicate(item))\n            break\n        list.add(item)\n    }\n    return list\n}\n\n/**\n *
Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.takeWhile(predicate: (UShort) -> Boolean): List<UShort> {\n    val list = ArrayList<UShort>()\n    for
(item in this) {\n        if (!predicate(item))\n            break\n        list.add(item)\n    }\n    return list\n}\n\n/**\n *
Reverses elements in the array in-place.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.reverse(): Unit {\n    storage.reverse()\n}\n\n/**\n *
Reverses elements in the array in-place.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.reverse(): Unit {\n    storage.reverse()\n}\n\n/**\n *
Reverses elements in the array in-place.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.reverse(): Unit {\n    storage.reverse()\n}\n\n/**\n *
Reverses elements in the array in-place.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.reverse(): Unit {\n    storage.reverse()\n}\n\n/**\n *
Reverses elements of the array in the specified
range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end
of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UIntArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    storage.reverse(fromIndex,
toIndex)\n}\n\n/**\n *
Reverses elements of the array in the specified range in-place.\n * \n * @param fromIndex
the start of the range (inclusive) to reverse.\n * @param toIndex the end of the range (exclusive) to reverse.\n *
\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of
this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    storage.reverse(fromIndex, toIndex)\n}\n\n/**\n *
Reverses elements of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range
(inclusive) to reverse.\n * @param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws

```

```

IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    storage.reverse(fromIndex, toIndex)\n}\n\n/**\n *
Reverses elements of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range
(inclusive) to reverse.\n * @param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    storage.reverse(fromIndex, toIndex)\n}\n\n/**\n *
Returns a list with elements in reversed order.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun UIntArray.reversed(): List<UInt> {\n    if (isEmpty()) return emptyList()\n    val list = toMutableList()\n
list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.reversed(): List<ULong> {\n
if (isEmpty()) return emptyList()\n    val list = toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n *
Returns a list with elements in reversed order.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UByteArray.reversed(): List<UByte> {\n    if (isEmpty()) return emptyList()\n    val list = toMutableList()\n
list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.reversed(): List<UShort> {\n
if (isEmpty()) return emptyList()\n    val list = toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n *
Returns an array with elements of this array in reversed order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.reversedArray(): UIntArray {\n    return UIntArray(storage.reversedArray())\n}\n\n/**\n * Returns an
array with elements of this array in reversed order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.reversedArray(): ULongArray {\n    return ULongArray(storage.reversedArray())\n}\n\n/**\n *
Returns an array with elements of this array in reversed order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.reversedArray(): UByteArray {\n    return UByteArray(storage.reversedArray())\n}\n\n/**\n * Returns
an array with elements of this array in reversed order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.reversedArray(): UShortArray {\n    return UShortArray(storage.reversedArray())\n}\n\n/**\n *
Randomly shuffles elements in this array in-place.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.shuffle(): Unit {\n
shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.shuffle(): Unit {\n
shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.shuffle(): Unit {\n
shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.shuffle(): Unit {\n
shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random]
instance as the source of randomness.\n * \n * See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.shuffle(random: Random): Unit
{\n    for (i in lastIndex downTo 1) {\n        val j = random.nextInt(i + 1)\n        val copy = this[i]\n        this[i] =
this[j]\n        this[j] = copy\n    }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified
[random] instance as the source of randomness.\n * \n * See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm\n

```



```

*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.shuffle(random: Random):
Unit {\n for (i in lastIndex downTo 1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] =
this[j]\n this[j] = copy\n }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified
[random] instance as the source of randomness.\n * \n * See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.shuffle(random: Random):
Unit {\n for (i in lastIndex downTo 1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] =
this[j]\n this[j] = copy\n }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified
[random] instance as the source of randomness.\n * \n * See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.shuffle(random: Random):
Unit {\n for (i in lastIndex downTo 1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] =
this[j]\n this[j] = copy\n }\n}\n\n/**\n * Sorts elements in the array in-place descending according to their
natural sort order.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UIntArray.sortDescending(): Unit {\n if (size > 1) {\n sort()\n reverse()\n }\n}\n\n/**\n * Sorts elements
in the array in-place descending according to their natural sort order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sortDescending(): Unit {\n if
(size > 1) {\n sort()\n reverse()\n }\n}\n\n/**\n * Sorts elements in the array in-place descending
according to their natural sort order.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UByteArray.sortDescending(): Unit {\n if (size > 1) {\n sort()\n reverse()\n }\n}\n\n/**\n * Sorts
elements in the array in-place descending according to their natural sort order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sortDescending(): Unit {\n if
(size > 1) {\n sort()\n reverse()\n }\n}\n\n/**\n * Returns a list of all elements sorted according to their
natural sort order.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sorted():
List<UInt> {\n return copyOf().apply { sort() }.asList()\n}\n\n/**\n * Returns a list of all elements sorted
according to their natural sort order.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
ULongArray.sorted(): List<ULong> {\n return copyOf().apply { sort() }.asList()\n}\n\n/**\n * Returns a list of all
elements sorted according to their natural sort order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sorted(): List<UByte> {\n
return copyOf().apply { sort() }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to their natural
sort order.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sorted():
List<UShort> {\n return copyOf().apply { sort() }.asList()\n}\n\n/**\n * Returns an array with all elements of this
array sorted according to their natural sort order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sortedArray(): UIntArray {\n if
(isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of
this array sorted according to their natural sort order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sortedArray(): ULongArray
{\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all
elements of this array sorted according to their natural sort order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sortedArray(): UByteArray {\n
if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements
of this array sorted according to their natural sort order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sortedArray(): UShortArray
{\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all
elements of this array sorted descending according to their natural sort order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sortedArrayDescending():
UIntArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns
an array with all elements of this array sorted descending according to their natural sort order.\n

```

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sortedArrayDescending():
ULongArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n *
Returns an array with all elements of this array sorted descending according to their natural sort order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sortedArrayDescending():
UByteArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n *
Returns an array with all elements of this array sorted descending according to their natural sort order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sortedArrayDescending():
UShortArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n *
Returns a list of all elements sorted descending according to their natural sort order.\n * \n * The sort is _stable_. It
means that equal elements preserve their order relative to each other after sorting.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sortedDescending(): List<UInt>
{\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending
according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order
relative to each other after sorting.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
ULongArray.sortedDescending(): List<ULong> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n *
Returns a list of all elements sorted descending according to their natural sort order.\n * \n * The sort is _stable_. It
means that equal elements preserve their order relative to each other after sorting.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sortedDescending():
List<UByte> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted
descending according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve
their order relative to each other after sorting.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun UShortArray.sortedDescending(): List<UShort> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n *
Returns an array of type [ByteArray], which is a view of this array where each element is a signed reinterpretation\n
* of the corresponding element of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.asByteArray(): ByteArray {\n return storage\n}\n\n/**\n * Returns an array of type [IntArray], which
is a view of this array where each element is a signed reinterpretation\n * of the corresponding element of this
array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.asIntArray(): IntArray {\n return storage\n}\n\n/**\n * Returns a [List] that wraps the original array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun UIntArray.asList():
List<UInt>\n\n/**\n * Returns a [List] that wraps the original array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun ULongArray.asList():
List<ULong>\n\n/**\n * Returns a [List] that wraps the original array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun UByteArray.asList():
List<UByte>\n\n/**\n * Returns a [List] that wraps the original array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun UShortArray.asList():
List<UShort>\n\n/**\n * Returns an array of type [LongArray], which is a view of this array where each element is
a signed reinterpretation\n * of the corresponding element of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.asLongArray(): LongArray {\n return storage\n}\n\n/**\n * Returns an array of type [ShortArray],
which is a view of this array where each element is a signed reinterpretation\n * of the corresponding element of this
array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.asShortArray(): ShortArray {\n return storage\n}\n\n/**\n * Returns an array of type [UByteArray],
which is a view of this array where each element is an unsigned reinterpretation\n * of the corresponding element of
this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun ByteArray.asUByteArray(): UByteArray {\n return UByteArray(this)\n}\n\n/**\n * Returns an array of type
[UIntArray], which is a view of this array where each element is an unsigned reinterpretation\n * of the
corresponding element of this array.\n

```

```

*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^@kotlin.internal.InlineOnly^public inline fun
IntArray.asUIntArray(): UIntArray {^n  return UIntArray(this)^n}^n/**^n * Returns an array of type
[UIntArray], which is a view of this array where each element is an unsigned reinterpretation^n * of the
corresponding element of this array.^n
*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^@kotlin.internal.InlineOnly^public inline fun
LongArray.asULongArray(): ULongArray {^n  return ULongArray(this)^n}^n/**^n * Returns an array of type
[UShortArray], which is a view of this array where each element is an unsigned reinterpretation^n * of the
corresponding element of this array.^n
*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^@kotlin.internal.InlineOnly^public inline fun
ShortArray.asUShortArray(): UShortArray {^n  return UShortArray(this)^n}^n/**^n * Returns `true` if the two
specified arrays are *structurally* equal to one another,^n * i.e. contain the same number of the same elements in the
same order.^n *^@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")^@SinceKotlin("1.3")^@DeprecatedSinceKotlin(hiddenSince =
"1.4")^@ExperimentalUnsignedTypes^public infix fun UIntArray.contentEquals(other: UIntArray): Boolean {^n
return this.contentEquals(other)^n}^n/**^n * Returns `true` if the two specified arrays are *structurally* equal to
one another,^n * i.e. contain the same number of the same elements in the same order.^n *^@Deprecated("Use
Kotlin compiler 1.4 to avoid deprecation
warning.")^@SinceKotlin("1.3")^@DeprecatedSinceKotlin(hiddenSince =
"1.4")^@ExperimentalUnsignedTypes^public infix fun ULongArray.contentEquals(other: ULongArray):
Boolean {^n  return this.contentEquals(other)^n}^n/**^n * Returns `true` if the two specified arrays are
*structurally* equal to one another,^n * i.e. contain the same number of the same elements in the same order.^n
*^@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")^@SinceKotlin("1.3")^@DeprecatedSinceKotlin(hiddenSince =
"1.4")^@ExperimentalUnsignedTypes^public infix fun UByteArray.contentEquals(other: UByteArray): Boolean
{^n  return this.contentEquals(other)^n}^n/**^n * Returns `true` if the two specified arrays are *structurally* equal
to one another,^n * i.e. contain the same number of the same elements in the same order.^n *^@Deprecated("Use
Kotlin compiler 1.4 to avoid deprecation
warning.")^@SinceKotlin("1.3")^@DeprecatedSinceKotlin(hiddenSince =
"1.4")^@ExperimentalUnsignedTypes^public infix fun UShortArray.contentEquals(other: UShortArray):
Boolean {^n  return this.contentEquals(other)^n}^n/**^n * Returns `true` if the two specified arrays are
*structurally* equal to one another,^n * i.e. contain the same number of the same elements in the same order.^n
*^@SinceKotlin("1.4")^@ExperimentalUnsignedTypes^public infix fun UIntArray?.contentEquals(other:
UIntArray?): Boolean {^n  return this?.storage.contentEquals(other?.storage)^n}^n/**^n * Returns `true` if the two
specified arrays are *structurally* equal to one another,^n * i.e. contain the same number of the same elements in the
same order.^n *^@SinceKotlin("1.4")^@ExperimentalUnsignedTypes^public infix fun
ULongArray?.contentEquals(other: ULongArray?): Boolean {^n  return
this?.storage.contentEquals(other?.storage)^n}^n/**^n * Returns `true` if the two specified arrays are *structurally*
equal to one another,^n * i.e. contain the same number of the same elements in the same order.^n
*^@SinceKotlin("1.4")^@ExperimentalUnsignedTypes^public infix fun UByteArray?.contentEquals(other:
UByteArray?): Boolean {^n  return this?.storage.contentEquals(other?.storage)^n}^n/**^n * Returns `true` if the
two specified arrays are *structurally* equal to one another,^n * i.e. contain the same number of the same elements
in the same order.^n *^@SinceKotlin("1.4")^@ExperimentalUnsignedTypes^public infix fun
UShortArray?.contentEquals(other: UShortArray?): Boolean {^n  return
this?.storage.contentEquals(other?.storage)^n}^n/**^n * Returns a hash code based on the contents of this array as
if it is [List].^n *^@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")^@SinceKotlin("1.3")^@DeprecatedSinceKotlin(hiddenSince =
"1.4")^@ExperimentalUnsignedTypes^public fun UIntArray.contentHashCode(): Int {^n  return
this.contentHashCode()^n}^n/**^n * Returns a hash code based on the contents of this array as if it is [List].^n

```

```

*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.\")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
"1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.contentHashCode(): Int {\n    return
this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n
*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.\")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
"1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.contentHashCode(): Int {\n    return
this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n
*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.\")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
"1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.contentHashCode(): Int {\n    return
this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray?.contentHashCode(): Int {\n
return this?.storage.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it
is [List].\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
ULongArray?.contentHashCode(): Int {\n    return this?.storage.contentHashCode()\n}\n\n/**\n * Returns a hash
code based on the contents of this array as if it is [List].\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray?.contentHashCode(): Int {\n
return this?.storage.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it
is [List].\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
UShortArray?.contentHashCode(): Int {\n    return this?.storage.contentHashCode()\n}\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n
\n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n
*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.\")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
"1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.contentToString(): String {\n    return
this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n
\n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*\n@Deprecated("Use
Kotlin compiler 1.4 to avoid deprecation
warning.\")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
"1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.contentToString(): String {\n    return
this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n
\n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*\n@Deprecated("Use
Kotlin compiler 1.4 to avoid deprecation
warning.\")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
"1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.contentToString(): String {\n    return
this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n
\n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*\n@Deprecated("Use
Kotlin compiler 1.4 to avoid deprecation
warning.\")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
"1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.contentToString(): String {\n    return
this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n
\n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray?.contentToString(): String {\n
return this?.joinToString(", ", "[", "]") ?: "null"\n}\n\n/**\n * Returns a string representation of the contents of
the specified array as if it is [List].\n
\n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray?.contentToString(): String {\n
return this?.joinToString(", ", "[", "]") ?: "null"\n}\n\n/**\n * Returns a string representation of the contents of

```

the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray?.contentToString(): String {\n
return this?.joinToString(", ", "[", "]") ?: "null"\n}\n\n/**\n * Returns a string representation of the contents of
the specified array as if it is [List].\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray?.contentToString(): String {\n
return this?.joinToString(", ", "[", "]") ?: "null"\n}\n\n/**\n * Copies this array or its subrange into the
[destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even
specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy
to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param
startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive)
of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or
[IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex
> endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array
starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n
*\n * @return the [destination] array.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.copyInto(destination: UIntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size):
UIntArray {\n
storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)\n
return
destination}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n *
It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the
destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the
[destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy,
0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n *
@throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of
range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the
subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index
is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.copyInto(destination: ULongArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size):
ULongArray {\n
storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)\n
return
destination}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n *
It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the
destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the
[destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy,
0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n *
@throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of
range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the
subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index
is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.copyInto(destination: UByteArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size):
UByteArray {\n
storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)\n
return
destination}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n *
It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the
destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the
[destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy,
```

0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.copyInto(destination: UShortArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int =
size): UShortArray {\n    storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)\n    return
destination\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.copyOfOf\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.copyOf(): UIntArray {\n    return UIntArray(storage.copyOf())\n}\n\n/**\n * Returns new array which is
a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.copyOf(): ULongArray {\n    return ULongArray(storage.copyOf())\n}\n\n/**\n * Returns new array
which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.copyOf(): UByteArray {\n    return UByteArray(storage.copyOf())\n}\n\n/**\n * Returns new array
which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.copyOf(): UShortArray {\n    return UShortArray(storage.copyOf())\n}\n\n/**\n * Returns new array
which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded
at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array
is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the
copy array are filled with zero values.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.copyOf(newSize: Int): UIntArray {\n    return UIntArray(storage.copyOf(newSize))\n}\n\n/**\n *
Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either
truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the
original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original
array, the extra elements in the copy array are filled with zero values.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.copyOf(newSize: Int): ULongArray {\n    return ULongArray(storage.copyOf(newSize))\n}\n\n/**\n *
Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either
truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the
original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original
array, the extra elements in the copy array are filled with zero values.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.copyOf(newSize: Int): UByteArray {\n    return UByteArray(storage.copyOf(newSize))\n}\n\n/**\n *
Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either
truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the
original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original
array, the extra elements in the copy array are filled with zero values.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.copyOf(newSize: Int): UShortArray {\n    return UShortArray(storage.copyOf(newSize))\n}\n\n/**\n *
Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start
of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *

```

```

@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.copyOfRange(fromIndex: Int, toIndex: Int): UIntArray {\n    return
UIntArray(storage.copyOfRange(fromIndex, toIndex))\n}\n\n/**\n * Returns a new array which is a copy of the
specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n *
@param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.copyOfRange(fromIndex: Int, toIndex: Int): ULongArray {\n    return
ULongArray(storage.copyOfRange(fromIndex, toIndex))\n}\n\n/**\n * Returns a new array which is a copy of the
specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n *
@param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.copyOfRange(fromIndex: Int, toIndex: Int): UByteArray {\n    return
UByteArray(storage.copyOfRange(fromIndex, toIndex))\n}\n\n/**\n * Returns a new array which is a copy of the
specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n *
@param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.copyOfRange(fromIndex: Int, toIndex: Int): UShortArray {\n    return
UShortArray(storage.copyOfRange(fromIndex, toIndex))\n}\n\n/**\n * Fills this array or its subrange with the
specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n *
@param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.fill(element: UInt, fromIndex:
Int = 0, toIndex: Int = size): Unit {\n    storage.fill(element.toInt(), fromIndex, toIndex)\n}\n\n/**\n * Fills this array
or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to
fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n *
@throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this
array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.fill(element: ULong,
fromIndex: Int = 0, toIndex: Int = size): Unit {\n    storage.fill(element.toLong(), fromIndex, toIndex)\n}\n\n/**\n *
Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range
(inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by
default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than
the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.fill(element: UByte,
fromIndex: Int = 0, toIndex: Int = size): Unit {\n    storage.fill(element.toByte(), fromIndex, toIndex)\n}\n\n/**\n *
Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range
(inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by
default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than
the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.fill(element: UShort,

```

```

fromIndex: Int = 0, toIndex: Int = size): Unit {
    storage.fill(element.toShort(), fromIndex, toIndex)
}

Returns the range of valid indices for the array.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public inline val UIntArray.indices: IntRange
    get() = storage.indices

Returns the range of valid indices for the array.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public inline val ULongArray.indices: IntRange
    get() = storage.indices

Returns the range of valid indices for the array.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public inline val UByteArray.indices: IntRange
    get() = storage.indices

Returns the range of valid indices for the array.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public inline val UShortArray.indices: IntRange
    get() = storage.indices

Returns the last valid index for the array.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public inline val UIntArray.lastIndex: Int
    get() = storage.lastIndex

Returns the last valid index for the array.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public inline val ULongArray.lastIndex: Int
    get() = storage.lastIndex

Returns the last valid index for the array.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public inline val UByteArray.lastIndex: Int
    get() = storage.lastIndex

Returns the last valid index for the array.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public inline val UShortArray.lastIndex: Int
    get() = storage.lastIndex

Returns an array containing all elements of the original array and then the given
[element].

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline
operator fun UIntArray.plus(element: UInt): UIntArray {
    return UIntArray(storage +
        element.toInt())

Returns an array containing all elements of the original array and then the given
[element].

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline
operator fun ULongArray.plus(element: ULong): ULongArray {
    return ULongArray(storage +
        element.toLong())

Returns an array containing all elements of the original array and then the given
[element].

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline
operator fun UByteArray.plus(element: UByte): UByteArray {
    return UByteArray(storage +
        element.toByte())

Returns an array containing all elements of the original array and then the given
[element].

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline
operator fun UShortArray.plus(element: UShort): UShortArray {
    return UShortArray(storage +
        element.toShort())

Returns an array containing all elements of the original array and then all elements
of the given [elements] collection.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public operator
fun UIntArray.plus(elements: Collection<UInt>): UIntArray {
    var index = size
    val result =
        storage.copyOf(size + elements.size)
    for (element in elements) result[index++] = element.toInt()
    return
        UIntArray(result)

Returns an array containing all elements of the original array and then all elements
of the given [elements] collection.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public operator
fun ULongArray.plus(elements: Collection<ULong>): ULongArray {
    var index = size
    val result =
        storage.copyOf(size + elements.size)
    for (element in elements) result[index++] = element.toLong()
    return
        ULongArray(result)

Returns an array containing all elements of the original array and then all
elements of the given [elements] collection.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public
operator fun UByteArray.plus(elements: Collection<UByte>): UByteArray {
    var index = size
    val result =
        storage.copyOf(size + elements.size)
    for (element in elements) result[index++] = element.toByte()
    return
        UByteArray(result)

Returns an array containing all elements of the original array and then all elements
of the given [elements] collection.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public operator
fun UShortArray.plus(elements: Collection<UShort>): UShortArray {
    var index = size
    val result =
        storage.copyOf(size + elements.size)
    for (element in elements) result[index++] = element.toShort()
    return
        UShortArray(result)

Returns an array containing all elements of the original array and then all
elements of the given [elements] array.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun

```



```

*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sort(fromIndex: Int = 0,
toIndex: Int = size): Unit {\n  AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  sortArray(this,
fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are
sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive)
to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException
if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentOutOfRangeException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sortDescending(fromIndex: Int,
toIndex: Int): Unit {\n  sort(fromIndex, toIndex)\n  reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of
the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort
order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range
(exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is
greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sortDescending(fromIndex:
Int, toIndex: Int): Unit {\n  sort(fromIndex, toIndex)\n  reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements
of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort
order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range
(exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is
greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sortDescending(fromIndex:
Int, toIndex: Int): Unit {\n  sort(fromIndex, toIndex)\n  reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements
of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort
order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range
(exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is
greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sortDescending(fromIndex:
Int, toIndex: Int): Unit {\n  sort(fromIndex, toIndex)\n  reverse(fromIndex, toIndex)\n}\n\n/**\n * Returns an
array of type [ByteArray], which is a copy of this array where each element is a signed reinterpretation\n * of the
corresponding element of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.toByteArray(): ByteArray {\n  return storage.copyOf()\n}\n\n/**\n * Returns an array of type
[IntArray], which is a copy of this array where each element is a signed reinterpretation\n * of the corresponding
element of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.toIntArray(): IntArray {\n  return storage.copyOf()\n}\n\n/**\n * Returns an array of type [LongArray],
which is a copy of this array where each element is a signed reinterpretation\n * of the corresponding element of this
array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.toLongArray(): LongArray {\n  return storage.copyOf()\n}\n\n/**\n * Returns an array of type
[ShortArray], which is a copy of this array where each element is a signed reinterpretation\n * of the corresponding
element of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.toShortArray(): ShortArray {\n  return storage.copyOf()\n}\n\n/**\n * Returns a *typed* object array
containing all of the elements of this primitive array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.toTypedArray(): Array<UInt>
{\n  return Array(size) { index -> this[index] }\n}\n\n/**\n * Returns a *typed* object array containing all of the
elements of this primitive array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
ULongArray.toTypedArray(): Array<ULong> {\n  return Array(size) { index -> this[index] }\n}\n\n/**\n *
Returns a *typed* object array containing all of the elements of this primitive array.\n

```

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.toTypedArray():
Array<UByte> {\n    return Array(size) { index -> this[index] }\n}\n\n/**\n * Returns a *typed* object array
containing all of the elements of this primitive array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.toTypedArray():
Array<UShort> {\n    return Array(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of UByte containing
all of the elements of this generic array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
Array<out UByte>.toUByteArray(): UByteArray {\n    return UByteArray(size) { index -> this[index] }\n}\n\n/**\n
 * Returns an array of type [UByteArray], which is a copy of this array where each element is an unsigned
reinterpretation\n * of the corresponding element of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ByteArray.toUByteArray(): UByteArray {\n    return UByteArray(this.copyOf())\n}\n\n/**\n * Returns an array of
UInt containing all of the elements of this generic array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Array<out UInt>.toUIntArray(): UIntArray
{\n    return UIntArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of type [UIntArray], which is a
copy of this array where each element is an unsigned reinterpretation\n * of the corresponding element of this
array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
IntArray.toUIntArray(): UIntArray {\n    return UIntArray(this.copyOf())\n}\n\n/**\n * Returns an array of ULong
containing all of the elements of this generic array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Array<out ULong>.toULongArray():
ULongArray {\n    return ULongArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of type
[ULongArray], which is a copy of this array where each element is an unsigned reinterpretation\n * of the
corresponding element of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
LongArray.toULongArray(): ULongArray {\n    return ULongArray(this.copyOf())\n}\n\n/**\n * Returns an array
of UShort containing all of the elements of this generic array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Array<out UShort>.toUShortArray():
UShortArray {\n    return UShortArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of type
[UShortArray], which is a copy of this array where each element is an unsigned reinterpretation\n * of the
corresponding element of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ShortArray.toUShortArray(): UShortArray {\n    return UShortArray(this.copyOf())\n}\n\n/**\n * Returns a [Map]
where keys are elements from the given array and values are\n * produced by the [valueSelector] function applied to
each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The returned map
preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Collections.Transformations.associateWith\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V>
UIntArray.associateWith(valueSelector: (UInt) -> V): Map<UInt, V> {\n    val result = LinkedHashMap<UInt,
V>(mapCapacity(size).coerceAtLeast(16))\n    return associateWithTo(result, valueSelector)\n}\n\n/**\n * Returns a
[Map] where keys are elements from the given array and values are\n * produced by the [valueSelector] function
applied to each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The
returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Collections.Transformations.associateWith\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V>
ULongArray.associateWith(valueSelector: (ULong) -> V): Map<ULong, V> {\n    val result =
LinkedHashMap<ULong, V>(mapCapacity(size).coerceAtLeast(16))\n    return associateWithTo(result,
valueSelector)\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n *
produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the last one
gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n
*

```

```

@sample samples.collections.Collections.Transformations.associateWith\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V>\nUByteArray.associateWith(valueSelector: (UByte) -> V): Map<UByte, V> {\n    val result =\n    LinkedHashMap<UByte, V>(mapCapacity(size).coerceAtLeast(16))\n    return associateWithTo(result,\n    valueSelector)\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n * produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the last one\n * gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Collections.Transformations.associateWith\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V>\nUShortArray.associateWith(valueSelector: (UShort) -> V): Map<UShort, V> {\n    val result =\n    LinkedHashMap<UShort, V>(mapCapacity(size).coerceAtLeast(16))\n    return associateWithTo(result,\n    valueSelector)\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each\n * element of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function\n * applied to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n * @sample samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in UInt, in V>> UIntArray.associateWithTo(destination: M, valueSelector: (UInt) -> V): M {\n    for\n    (element in this) {\n        destination.put(element, valueSelector(element))\n    }\n    return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n * @sample\n    samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in ULong, in V>> ULongArray.associateWithTo(destination: M, valueSelector: (ULong) -> V): M {\n    for\n    (element in this) {\n        destination.put(element, valueSelector(element))\n    }\n    return\n    destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each element\n * of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function applied\n * to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n * @sample\n    samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in UByte, in V>> UByteArray.associateWithTo(destination: M, valueSelector: (UByte) -> V): M {\n    for\n    (element in this) {\n        destination.put(element, valueSelector(element))\n    }\n    return\n    destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each element\n * of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function applied\n * to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n * @sample\n    samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in UShort, in V>> UShortArray.associateWithTo(destination: M, valueSelector: (UShort) -> V): M {\n    for\n    (element in this) {\n        destination.put(element, valueSelector(element))\n    }\n    return\n    destination\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being\n * invoked on each element of original array.\n * \n * @sample\n    samples.collections.Collections.Transformations.flatMap\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nUIntArray.flatMap(transform: (UInt) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(),\n    transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being\n * invoked on each element of original array.\n * \n * @sample\n    samples.collections.Collections.Transformations.flatMap\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>

```

```

ULongArray.flatMap(transform: (ULong) -> Iterable<R>): List<R> {\n  return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.flatMap(transform: (UByte) -> Iterable<R>): List<R> {\n  return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.flatMap(transform: (UShort) -> Iterable<R>): List<R> {\n  return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element\n * and its index in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.flatMapIndexed(transform: (index: Int, UInt) -> Iterable<R>): List<R> {\n  return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.flatMapIndexed(transform: (index: Int, ULong) -> Iterable<R>): List<R> {\n  return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.flatMapIndexed(transform: (index: Int, UByte) -> Iterable<R>): List<R> {\n  return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.flatMapIndexed(transform: (index: Int, UShort) -> Iterable<R>): List<R> {\n  return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UIntArray.flatMapIndexedTo(destination: C, transform: (index: Int, UInt) ->
Iterable<R>): C {\n  var index = 0\n  for (element in this) {\n    val list = transform(index++, element)\n
destination.addAll(list)\n  }\n  return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> ULongArray.flatMapIndexedTo(destination: C, transform: (index: Int, ULong) ->

```

```

Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}
Append all elements yielded from results of [transform] function being invoked on each element and its index in the original array, to the given [destination].

*SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolutionByLambdaReturnType@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> UByteArray.flatMapIndexedTo(destination: C, transform: (index: Int, UByte) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}
Append all elements yielded from results of [transform] function being invoked on each element and its index in the original array, to the given [destination].

*SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolutionByLambdaReturnType@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> UShortArray.flatMapIndexedTo(destination: C, transform: (index: Int, UShort) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}
Append all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].

*SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> UIntArray.flatMapTo(destination: C, transform: (UInt) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}
Append all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].

*SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> ULongArray.flatMapTo(destination: C, transform: (ULong) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}
Append all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].

*SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> UByteArray.flatMapTo(destination: C, transform: (UByte) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}
Append all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].

*SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> UShortArray.flatMapTo(destination: C, transform: (UShort) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}
Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and returns a map where each group key is associated with a list of corresponding elements.
The returned map preserves the entry iteration order of the keys produced from the original array.
@sample samples.collections.Collections.Transformations.groupBy

*SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <K> UIntArray.groupBy(keySelector: (UInt) -> K): Map<K, List<UInt>> {
    return groupByTo(LinkedHashMap<K, MutableList<UInt>>(), keySelector)
}
Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and returns a map where each group key is associated with a list of corresponding elements.
The returned map preserves the entry iteration order of the keys produced from the original array.
@sample samples.collections.Collections.Transformations.groupBy

*SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <K> ULongArray.groupBy(keySelector: (ULong) -> K): Map<K, List<ULong>> {
    return groupByTo(LinkedHashMap<K, MutableList<ULong>>(), keySelector)
}
Groups elements of the

```

original array by the key returned by the given [keySelector] function
* applied to each element and returns a map where each group key is associated with a list of corresponding elements.
* The returned map preserves the entry iteration order of the keys produced from the original array.
* @sample

samples.collections.Collections.Transformations.groupBy

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K>\nUByteArray.groupBy(keySelector: (UByte) -> K): Map<K, List<UByte>> {\n    return
```

groupByTo(LinkedHashMap<K, MutableList<UByte>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function
* applied to each element and returns a map where each group key is associated with a list of corresponding elements.
* The returned map preserves the entry iteration order of the keys produced from the original array.
* @sample

samples.collections.Collections.Transformations.groupBy

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K>\nUShortArray.groupBy(keySelector: (UShort) -> K): Map<K, List<UShort>> {\n    return
```

groupByTo(LinkedHashMap<K, MutableList<UShort>>(), keySelector)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array
* by the key returned by the given [keySelector] function applied to the element
* and returns a map where each group key is associated with a list of corresponding values.
* The returned map preserves the entry iteration order of the keys produced from the original array.
* @sample

samples.collections.Collections.Transformations.groupByKeysAndValues

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V>\nUIntArray.groupBy(keySelector: (UInt) -> K, valueTransform: (UInt) -> V): Map<K, List<V>> {\n    return
```

groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array
* by the key returned by the given [keySelector] function applied to the element
* and returns a map where each group key is associated with a list of corresponding values.
* The returned map preserves the entry iteration order of the keys produced from the original array.
* @sample

samples.collections.Collections.Transformations.groupByKeysAndValues

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V>\nULongArray.groupBy(keySelector: (ULong) -> K, valueTransform: (ULong) -> V): Map<K, List<V>> {\n    return
```

groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array
* by the key returned by the given [keySelector] function applied to the element
* and returns a map where each group key is associated with a list of corresponding values.
* The returned map preserves the entry iteration order of the keys produced from the original array.
* @sample

samples.collections.Collections.Transformations.groupByKeysAndValues

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V>\nUByteArray.groupBy(keySelector: (UByte) -> K, valueTransform: (UByte) -> V): Map<K, List<V>> {\n    return
```

groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array
* by the key returned by the given [keySelector] function applied to the element
* and returns a map where each group key is associated with a list of corresponding values.
* The returned map preserves the entry iteration order of the keys produced from the original array.
* @sample

samples.collections.Collections.Transformations.groupByKeysAndValues

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V>\nUShortArray.groupBy(keySelector: (UShort) -> K, valueTransform: (UShort) -> V): Map<K, List<V>> {\n    return
```

groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function
* applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.
* The returned map preserves the entry iteration order of the keys produced from the original array.
* @return The [destination] map.
* @sample

samples.collections.Collections.Transformations.groupBy

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, M :
MutableMap<in K, MutableList<UInt>>> UIntArray.groupByTo(destination: M, keySelector: (UInt) -> K): M {\n
for (element in this) {\n    val key = keySelector(element)\n    val list = destination.getOrPut(key) {
ArrayList<UInt>() }\n    list.add(element)\n } \n return destination\n}\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the
[destination] map each group key associated with a list of corresponding elements.\n * \n * @return The
[destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, M :
MutableMap<in K, MutableList<ULong>>> ULongArray.groupByTo(destination: M, keySelector: (ULong) -> K):
M {\n for (element in this) {\n    val key = keySelector(element)\n    val list = destination.getOrPut(key) {
ArrayList<ULong>() }\n    list.add(element)\n } \n return destination\n}\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the
[destination] map each group key associated with a list of corresponding elements.\n * \n * @return The
[destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, M :
MutableMap<in K, MutableList<UByte>>> UByteArray.groupByTo(destination: M, keySelector: (UByte) -> K):
M {\n for (element in this) {\n    val key = keySelector(element)\n    val list = destination.getOrPut(key) {
ArrayList<UByte>() }\n    list.add(element)\n } \n return destination\n}\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the
[destination] map each group key associated with a list of corresponding elements.\n * \n * @return The
[destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, M :
MutableMap<in K, MutableList<UShort>>> UShortArray.groupByTo(destination: M, keySelector: (UShort) -> K):
M {\n for (element in this) {\n    val key = keySelector(element)\n    val list = destination.getOrPut(key) {
ArrayList<UShort>() }\n    list.add(element)\n } \n return destination\n}\n\n/**\n * Groups values returned by
the [valueTransform] function applied to each element of the original array\n * by the key returned by the given
[keySelector] function applied to the element\n * and puts to the [destination] map each group key associated with a
list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V,
M : MutableMap<in K, MutableList<V>>> UIntArray.groupByTo(destination: M, keySelector: (UInt) -> K,
valueTransform: (UInt) -> V): M {\n for (element in this) {\n    val key = keySelector(element)\n    val list =
destination.getOrPut(key) { ArrayList<V>() }\n    list.add(valueTransform(element))\n } \n return
destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the
original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the
[destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination]
map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V,
M : MutableMap<in K, MutableList<V>>> ULongArray.groupByTo(destination: M, keySelector: (ULong) -> K,
valueTransform: (ULong) -> V): M {\n for (element in this) {\n    val key = keySelector(element)\n    val list =
destination.getOrPut(key) { ArrayList<V>() }\n    list.add(valueTransform(element))\n } \n return
destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the
original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the
[destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination]
map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V,
M : MutableMap<in K, MutableList<V>>> UByteArray.groupByTo(destination: M, keySelector: (UByte) -> K,
valueTransform: (UByte) -> V): M {\n for (element in this) {\n    val key = keySelector(element)\n    val list

```


transform applied to the element.\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :  
MutableCollection<in R>> UIntArray.mapIndexedTo(destination: C, transform: (index: Int, UInt) -> R): C {\n    var index = 0\n    for (item in this)\n        destination.add(transform(index++, item))\n    return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element  
itself\n * and returns the result of the transform applied to the element.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :  
MutableCollection<in R>> ULongArray.mapIndexedTo(destination: C, transform: (index: Int, ULong) -> R): C {\n    var index = 0\n    for (item in this)\n        destination.add(transform(index++, item))\n    return  
destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an  
element and the element itself\n * and returns the result of the transform applied to the element.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :  
MutableCollection<in R>> UByteArray.mapIndexedTo(destination: C, transform: (index: Int, UByte) -> R): C {\n    var index = 0\n    for (item in this)\n        destination.add(transform(index++, item))\n    return  
destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an  
element and the element itself\n * and returns the result of the transform applied to the element.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :  
MutableCollection<in R>> UShortArray.mapIndexedTo(destination: C, transform: (index: Int, UShort) -> R): C {\n    var index = 0\n    for (item in this)\n        destination.add(transform(index++, item))\n    return  
destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :  
MutableCollection<in R>> UIntArray.mapTo(destination: C, transform: (UInt) -> R): C {\n    for (item in this)\n        destination.add(transform(item))\n    return destination\n}\n\n/**\n * Applies the given [transform] function to each  
element of the original array\n * and appends the results to the given [destination].\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :  
MutableCollection<in R>> ULongArray.mapTo(destination: C, transform: (ULong) -> R): C {\n    for (item in  
this)\n        destination.add(transform(item))\n    return destination\n}\n\n/**\n * Applies the given [transform]  
function to each element of the original array\n * and appends the results to the given [destination].\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :  
MutableCollection<in R>> UByteArray.mapTo(destination: C, transform: (UByte) -> R): C {\n    for (item in this)\n        destination.add(transform(item))\n    return destination\n}\n\n/**\n * Applies the given [transform] function to  
each element of the original array\n * and appends the results to the given [destination].\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :  
MutableCollection<in R>> UShortArray.mapTo(destination: C, transform: (UShort) -> R): C {\n    for (item in  
this)\n        destination.add(transform(item))\n    return destination\n}\n\n/**\n * Returns a lazy [Iterable] that wraps  
each element of the original array\n * into an [IndexedValue] containing the index of that element and the element  
itself.\n * @SinceKotlin("1.3")\n * @ExperimentalUnsignedTypes\n * public fun UIntArray.withIndex():  
Iterable<IndexedValue<UInt>> {\n    return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable]  
that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the  
element itself.\n * @SinceKotlin("1.3")\n * @ExperimentalUnsignedTypes\n * public fun ULongArray.withIndex():  
Iterable<IndexedValue<ULong>> {\n    return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable]  
that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the  
element itself.\n * @SinceKotlin("1.3")\n * @ExperimentalUnsignedTypes\n * public fun UByteArray.withIndex():  
Iterable<IndexedValue<UByte>> {\n    return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable]
```

```
that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the  
element itself.\n * @SinceKotlin("1.3")\n * @ExperimentalUnsignedTypes\n * public fun ULongArray.withIndex():  
Iterable<IndexedValue<ULong>> {\n    return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable]  
that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the  
element itself.\n * @SinceKotlin("1.3")\n * @ExperimentalUnsignedTypes\n * public fun UByteArray.withIndex():  
Iterable<IndexedValue<UByte>> {\n    return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable]
```

that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

```

*\/n@SinceKotlin("1.3")\/n@ExperimentalUnsignedTypes\/npublic fun UShortArray.withIndex():
Iterable<IndexedValue<UShort>> {\/n    return IndexingIterable { iterator() }\/n}\/n\/n**\/n * Returns `true` if all
elements match the given [predicate].\/n *\/n * @sample samples.collections.Collections.Aggregates.all\/n
*\/n@SinceKotlin("1.3")\/n@ExperimentalUnsignedTypes\/n@kotlin.internal.InlineOnly\/npublic inline fun
UIntArray.all(predicate: (UInt) -> Boolean): Boolean {\/n    for (element in this) if (!predicate(element)) return
false\/n    return true\/n}\/n}\/n**\/n * Returns `true` if all elements match the given [predicate].\/n *\/n * @sample
samples.collections.Collections.Aggregates.all\/n
*\/n@SinceKotlin("1.3")\/n@ExperimentalUnsignedTypes\/n@kotlin.internal.InlineOnly\/npublic inline fun
ULongArray.all(predicate: (ULong) -> Boolean): Boolean {\/n    for (element in this) if (!predicate(element)) return
false\/n    return true\/n}\/n}\/n**\/n * Returns `true` if all elements match the given [predicate].\/n *\/n * @sample
samples.collections.Collections.Aggregates.all\/n
*\/n@SinceKotlin("1.3")\/n@ExperimentalUnsignedTypes\/n@kotlin.internal.InlineOnly\/npublic inline fun
UByteArray.all(predicate: (UByte) -> Boolean): Boolean {\/n    for (element in this) if (!predicate(element)) return
false\/n    return true\/n}\/n}\/n**\/n * Returns `true` if all elements match the given [predicate].\/n *\/n * @sample
samples.collections.Collections.Aggregates.all\/n
*\/n@SinceKotlin("1.3")\/n@ExperimentalUnsignedTypes\/n@kotlin.internal.InlineOnly\/npublic inline fun
UShortArray.all(predicate: (UShort) -> Boolean): Boolean {\/n    for (element in this) if (!predicate(element)) return
false\/n    return true\/n}\/n}\/n**\/n * Returns `true` if array has at least one element.\/n *\/n * @sample
samples.collections.Collections.Aggregates.any\/n
*\/n@SinceKotlin("1.3")\/n@ExperimentalUnsignedTypes\/n@kotlin.internal.InlineOnly\/npublic inline fun
UIntArray.any(): Boolean {\/n    return storage.any()\/n}\/n}\/n**\/n * Returns `true` if array has at least one element.\/n
*\/n * @sample samples.collections.Collections.Aggregates.any\/n
*\/n@SinceKotlin("1.3")\/n@ExperimentalUnsignedTypes\/n@kotlin.internal.InlineOnly\/npublic inline fun
ULongArray.any(): Boolean {\/n    return storage.any()\/n}\/n}\/n**\/n * Returns `true` if array has at least one
element.\/n *\/n * @sample samples.collections.Collections.Aggregates.any\/n
*\/n@SinceKotlin("1.3")\/n@ExperimentalUnsignedTypes\/n@kotlin.internal.InlineOnly\/npublic inline fun
UByteArray.any(): Boolean {\/n    return storage.any()\/n}\/n}\/n**\/n * Returns `true` if array has at least one
element.\/n *\/n * @sample samples.collections.Collections.Aggregates.any\/n
*\/n@SinceKotlin("1.3")\/n@ExperimentalUnsignedTypes\/n@kotlin.internal.InlineOnly\/npublic inline fun
UShortArray.any(): Boolean {\/n    return storage.any()\/n}\/n}\/n**\/n * Returns `true` if at least one element matches
the given [predicate].\/n *\/n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\/n
*\/n@SinceKotlin("1.3")\/n@ExperimentalUnsignedTypes\/n@kotlin.internal.InlineOnly\/npublic inline fun
UIntArray.any(predicate: (UInt) -> Boolean): Boolean {\/n    for (element in this) if (predicate(element)) return true\/n
    return false\/n}\/n}\/n**\/n * Returns `true` if at least one element matches the given [predicate].\/n *\/n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\/n
*\/n@SinceKotlin("1.3")\/n@ExperimentalUnsignedTypes\/n@kotlin.internal.InlineOnly\/npublic inline fun
ULongArray.any(predicate: (ULong) -> Boolean): Boolean {\/n    for (element in this) if (predicate(element)) return
true\/n    return false\/n}\/n}\/n**\/n * Returns `true` if at least one element matches the given [predicate].\/n *\/n *
@sample samples.collections.Collections.Aggregates.anyWithPredicate\/n
*\/n@SinceKotlin("1.3")\/n@ExperimentalUnsignedTypes\/n@kotlin.internal.InlineOnly\/npublic inline fun
UByteArray.any(predicate: (UByte) -> Boolean): Boolean {\/n    for (element in this) if (predicate(element)) return
true\/n    return false\/n}\/n}\/n**\/n * Returns `true` if at least one element matches the given [predicate].\/n *\/n *
@sample samples.collections.Collections.Aggregates.anyWithPredicate\/n
*\/n@SinceKotlin("1.3")\/n@ExperimentalUnsignedTypes\/n@kotlin.internal.InlineOnly\/npublic inline fun
UShortArray.any(predicate: (UShort) -> Boolean): Boolean {\/n    for (element in this) if (predicate(element)) return
true\/n    return false\/n}\/n}\/n**\/n * Returns the number of elements matching the given [predicate].\/n
*\/n@SinceKotlin("1.3")\/n@ExperimentalUnsignedTypes\/n@kotlin.internal.InlineOnly\/npublic inline fun

```

```

UIntArray.count(predicate: (UInt) -> Boolean): Int { \n  var count = 0 \n  for (element in this) if
(predicate(element)) ++count \n  return count \n} \n \n /** \n * Returns the number of elements matching the given
[predicate]. \n * \n @SinceKotlin("1.3") \n @ExperimentalUnsignedTypes \n @kotlin.internal.InlineOnly \n public
inline fun ULongArray.count(predicate: (ULong) -> Boolean): Int { \n  var count = 0 \n  for (element in this) if
(predicate(element)) ++count \n  return count \n} \n \n /** \n * Returns the number of elements matching the given
[predicate]. \n * \n @SinceKotlin("1.3") \n @ExperimentalUnsignedTypes \n @kotlin.internal.InlineOnly \n public
inline fun UByteArray.count(predicate: (UByte) -> Boolean): Int { \n  var count = 0 \n  for (element in this) if
(predicate(element)) ++count \n  return count \n} \n \n /** \n * Returns the number of elements matching the given
[predicate]. \n * \n @SinceKotlin("1.3") \n @ExperimentalUnsignedTypes \n @kotlin.internal.InlineOnly \n public
inline fun UShortArray.count(predicate: (UShort) -> Boolean): Int { \n  var count = 0 \n  for (element in this) if
(predicate(element)) ++count \n  return count \n} \n \n /** \n * Accumulates value starting with [initial] value and
applying [operation] from left to right \n * to current accumulator value and each element. \n * \n * Returns the
specified [initial] value if the array is empty. \n * \n * @param [operation] function that takes current accumulator
value and an element, and calculates the next accumulator value. \n
*\n @SinceKotlin("1.3") \n @ExperimentalUnsignedTypes \n @kotlin.internal.InlineOnly \n public inline fun <R>
UIntArray.fold(initial: R, operation: (acc: R, UInt) -> R): R { \n  var accumulator = initial \n  for (element in this)
accumulator = operation(accumulator, element) \n  return accumulator \n} \n \n /** \n * Accumulates value starting
with [initial] value and applying [operation] from left to right \n * to current accumulator value and each element. \n *
Returns the specified [initial] value if the array is empty. \n * \n * @param [operation] function that takes
current accumulator value and an element, and calculates the next accumulator value. \n
*\n @SinceKotlin("1.3") \n @ExperimentalUnsignedTypes \n @kotlin.internal.InlineOnly \n public inline fun <R>
ULongArray.fold(initial: R, operation: (acc: R, ULong) -> R): R { \n  var accumulator = initial \n  for (element in
this) accumulator = operation(accumulator, element) \n  return accumulator \n} \n \n /** \n * Accumulates value
starting with [initial] value and applying [operation] from left to right \n * to current accumulator value and each
element. \n * \n * Returns the specified [initial] value if the array is empty. \n * \n * @param [operation] function that
takes current accumulator value and an element, and calculates the next accumulator value. \n
*\n @SinceKotlin("1.3") \n @ExperimentalUnsignedTypes \n @kotlin.internal.InlineOnly \n public inline fun <R>
UByteArray.fold(initial: R, operation: (acc: R, UByte) -> R): R { \n  var accumulator = initial \n  for (element in
this) accumulator = operation(accumulator, element) \n  return accumulator \n} \n \n /** \n * Accumulates value
starting with [initial] value and applying [operation] from left to right \n * to current accumulator value and each
element. \n * \n * Returns the specified [initial] value if the array is empty. \n * \n * @param [operation] function that
takes current accumulator value and an element, and calculates the next accumulator value. \n
*\n @SinceKotlin("1.3") \n @ExperimentalUnsignedTypes \n @kotlin.internal.InlineOnly \n public inline fun <R>
UShortArray.fold(initial: R, operation: (acc: R, UShort) -> R): R { \n  var accumulator = initial \n  for (element in
this) accumulator = operation(accumulator, element) \n  return accumulator \n} \n \n /** \n * Accumulates value
starting with [initial] value and applying [operation] from left to right \n * to current accumulator value and each
element with its index in the original array. \n * \n * Returns the specified [initial] value if the array is empty. \n *
\n * @param [operation] function that takes the index of an element, current accumulator value \n * and the element
itself, and calculates the next accumulator value. \n
*\n @SinceKotlin("1.3") \n @ExperimentalUnsignedTypes \n @kotlin.internal.InlineOnly \n public inline fun <R>
UIntArray.foldIndexed(initial: R, operation: (index: Int, acc: R, UInt) -> R): R { \n  var index = 0 \n  var
accumulator = initial \n  for (element in this) accumulator = operation(index++, accumulator, element) \n  return
accumulator \n} \n \n /** \n * Accumulates value starting with [initial] value and applying [operation] from left to
right \n * to current accumulator value and each element with its index in the original array. \n * \n * Returns the
specified [initial] value if the array is empty. \n * \n * @param [operation] function that takes the index of an
element, current accumulator value \n * and the element itself, and calculates the next accumulator value. \n
*\n @SinceKotlin("1.3") \n @ExperimentalUnsignedTypes \n @kotlin.internal.InlineOnly \n public inline fun <R>
ULongArray.foldIndexed(initial: R, operation: (index: Int, acc: R, ULong) -> R): R { \n  var index = 0 \n  var

```

```

accumulator = initial\n  for (element in this) accumulator = operation(index++, accumulator, element)\n  return
accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to
right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns the
specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an
element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.foldIndexed(initial: R, operation: (index: Int, acc: R, UByte) -> R): R {\n  var index = 0\n  var
accumulator = initial\n  for (element in this) accumulator = operation(index++, accumulator, element)\n  return
accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to
right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns the
specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an
element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.foldIndexed(initial: R, operation: (index: Int, acc: R, UShort) -> R): R {\n  var index = 0\n  var
accumulator = initial\n  for (element in this) accumulator = operation(index++, accumulator, element)\n  return
accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to
left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is
empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the
next accumulator value.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.foldRight(initial: R, operation: (UInt, acc: R) -> R): R {\n  var index = lastIndex\n  var accumulator =
initial\n  while (index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n  }\n  return
accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to
left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is
empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the
next accumulator value.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.foldRight(initial: R, operation: (ULong, acc: R) -> R): R {\n  var index = lastIndex\n  var
accumulator = initial\n  while (index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n  }\n
return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right
to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is
empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the
next accumulator value.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.foldRight(initial: R, operation: (UByte, acc: R) -> R): R {\n  var index = lastIndex\n  var
accumulator = initial\n  while (index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n  }\n
return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right
to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is
empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the
next accumulator value.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.foldRight(initial: R, operation: (UShort, acc: R) -> R): R {\n  var index = lastIndex\n  var
accumulator = initial\n  while (index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n  }\n
return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right
to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the
specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an
element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>

```

```

UIntArray.foldRightIndexed(initial: R, operation: (index: Int, UInt, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element with its index in the original array and current accumulator value.

Returns the specified [initial] value if the array is empty.

@param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R> ULongArray.foldRightIndexed(initial: R, operation: (index: Int, ULong, acc: R) -> R): R {
    var index =
    lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index),
        accumulator)
        --index
    }
    return accumulator
}

```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element with its index in the original array and current accumulator value.

Returns the specified [initial] value if the array is empty.

@param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R>
UByteArray.foldRightIndexed(initial: R, operation: (index: Int, UByte, acc: R) -> R): R {
    var index =
    lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index),
        accumulator)
        --index
    }
    return accumulator
}

```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element with its index in the original array and current accumulator value.

Returns the specified [initial] value if the array is empty.

@param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R>
UShortArray.foldRightIndexed(initial: R, operation: (index: Int, UShort, acc: R) -> R): R {
    var index =
    lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index),
        accumulator)
        --index
    }
    return accumulator
}

```

Performs the given [action] on each element.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun
UIntArray.forEach(action: (UInt) -> Unit): Unit {
    for (element in this) action(element)
}

```

Performs the given [action] on each element.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun
ULongArray.forEach(action: (ULong) -> Unit): Unit {
    for (element in this) action(element)
}

```

Performs the given [action] on each element.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun
UByteArray.forEach(action: (UByte) -> Unit): Unit {
    for (element in this) action(element)
}

```

Performs the given [action] on each element.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun
UShortArray.forEach(action: (UShort) -> Unit): Unit {
    for (element in this) action(element)
}

```

Performs the given [action] on each element, providing sequential index with the element.

@param [action] function that takes the index of an element and the element itself and performs the action on the element.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun
UIntArray.forEachIndexed(action: (index: Int, UInt) -> Unit): Unit {
    var index = 0
    for (item in this)
    action(index++, item)
}

```

Performs the given [action] on each element, providing sequential index with the element.

@param [action] function that takes the index of an element and the element itself and performs the action on the element.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun
ULongArray.forEachIndexed(action: (index: Int, ULong) -> Unit): Unit {
    var index = 0
    for (item in this)
    action(index++, item)
}

```

Performs the given [action] on each element, providing sequential index with the element.

@param [action] function that takes the index of an element and the element itself and performs

the action on the element.\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun  
UByteArray.forEachIndexed(action: (index: Int, UByte) -> Unit): Unit {\n    var index = 0\n    for (item in this)  
    action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with  
the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs  
the action on the element.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun  
UShortArray.forEachIndexed(action: (index: Int, UShort) -> Unit): Unit {\n    var index = 0\n    for (item in this)  
    action(index++, item)\n}\n\n/**\n * Returns the largest element.\n * \n * @throws NoSuchElementException if the  
array is empty.\n *\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow-  
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun  
UIntArray.max(): UInt {\n    if (isEmpty()) throw NoSuchElementException()\n    var max = this[0]\n    for (i in  
1..lastIndex) {\n        val e = this[i]\n        if (max < e) max = e\n    }\n    return max\n}\n\n/**\n * Returns the largest  
element.\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow-  
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun  
ULongArray.max(): ULong {\n    if (isEmpty()) throw NoSuchElementException()\n    var max = this[0]\n    for (i  
in 1..lastIndex) {\n        val e = this[i]\n        if (max < e) max = e\n    }\n    return max\n}\n\n/**\n * Returns the  
largest element.\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow-  
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun  
UByteArray.max(): UByte {\n    if (isEmpty()) throw NoSuchElementException()\n    var max = this[0]\n    for (i in  
1..lastIndex) {\n        val e = this[i]\n        if (max < e) max = e\n    }\n    return max\n}\n\n/**\n * Returns the largest  
element.\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxOrThrow-  
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun  
UShortArray.max(): UShort {\n    if (isEmpty()) throw NoSuchElementException()\n    var max = this[0]\n    for (i  
in 1..lastIndex) {\n        val e = this[i]\n        if (max < e) max = e\n    }\n    return max\n}\n\n/**\n * Returns the first  
element yielding the largest value of the given function.\n * \n * @throws NoSuchElementException if the array is  
empty.\n * \n * @sample samples.collections.Collections.Aggregates.maxBy\n
```

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow-  
U")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n@Suppress("CONFLICTING_OVERLOADS  
")\npublic inline fun <R : Comparable<R>> UIntArray.maxBy(selector: (UInt) -> R): UInt {\n    if (isEmpty())  
    throw NoSuchElementException()\n    var maxElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex ==  
0) return maxElem\n    var maxValue = selector(maxElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val  
v = selector(e)\n        if (maxValue < v) {\n            maxElem = e\n            maxValue = v\n        }\n    }\n    return  
maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n * @throws  
NoSuchElementException if the array is empty.\n * \n * @sample  
samples.collections.Collections.Aggregates.maxBy\n
```

```
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxByOrThrow-  
U")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n@Suppress("CONFLICTING_OVERLOADS  
")\npublic inline fun <R : Comparable<R>> ULongArray.maxBy(selector: (ULong) -> R): ULong {\n    if  
(isEmpty()) throw NoSuchElementException()\n    var maxElem = this[0]\n    val lastIndex = this.lastIndex\n    if  
(lastIndex == 0) return maxElem\n    var maxValue = selector(maxElem)\n    for (i in 1..lastIndex) {\n        val e =  
this[i]\n        val v = selector(e)\n        if (maxValue < v) {\n            maxElem = e\n            maxValue = v\n        }\n    }\n    return  
maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n * @throws  
NoSuchElementException if the array is empty.\n * \n * @sample  
samples.collections.Collections.Aggregates.maxBy\n
```

```

*\/n@SinceKotlin("1.7")n@kotlin.jvm.JvmName("maxByOrThrow-
U")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlyn@Suppress("CONFLICTING_OVERLOADS
")npublic inline fun <R : Comparable<R>> UByteArray.maxBy(selector: (UByte) -> R): UByte {n if
(isEmpty()) throw NoSuchElementException()n var maxElem = this[0]n val lastIndex = this.lastIndexn if
(lastIndex == 0) return maxElemn var maxValue = selector(maxElem)n for (i in 1..lastIndex) {n val e =
this[i]n val v = selector(e)n if (maxValue < v) {n maxElem = e\n maxValue = v\n }n
}n return maxElem}n/n/**n * Returns the first element yielding the largest value of the given function.n *
* @throws NoSuchElementException if the array is empty.n * n * @sample
samples.collections.Collections.Aggregates.maxByn
*\/n@SinceKotlin("1.7")n@kotlin.jvm.JvmName("maxByOrThrow-
U")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlyn@Suppress("CONFLICTING_OVERLOADS
")npublic inline fun <R : Comparable<R>> UShortArray.maxBy(selector: (UShort) -> R): UShort {n if
(isEmpty()) throw NoSuchElementException()n var maxElem = this[0]n val lastIndex = this.lastIndexn if
(lastIndex == 0) return maxElemn var maxValue = selector(maxElem)n for (i in 1..lastIndex) {n val e =
this[i]n val v = selector(e)n if (maxValue < v) {n maxElem = e\n maxValue = v\n }n
}n return maxElem}n/n/**n * Returns the first element yielding the largest value of the given function or
`null` if there are no elements.n * n * @sample samples.collections.Collections.Aggregates.maxByOrNulln
*\/n@SinceKotlin("1.4")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun <R :
Comparable<R>> UIntArray.maxByOrNull(selector: (UInt) -> R): UInt? {n if (isEmpty()) return nulln var
maxElem = this[0]n val lastIndex = this.lastIndexn if (lastIndex == 0) return maxElemn var maxValue =
selector(maxElem)n for (i in 1..lastIndex) {n val e = this[i]n val v = selector(e)n if (maxValue < v)
{n maxElem = e\n maxValue = v\n }n }n return maxElem}n/n/**n * Returns the first
element yielding the largest value of the given function or `null` if there are no elements.n * n * @sample
samples.collections.Collections.Aggregates.maxByOrNulln
*\/n@SinceKotlin("1.4")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun <R :
Comparable<R>> ULongArray.maxByOrNull(selector: (ULong) -> R): ULong? {n if (isEmpty()) return nulln
var maxElem = this[0]n val lastIndex = this.lastIndexn if (lastIndex == 0) return maxElemn var maxValue =
selector(maxElem)n for (i in 1..lastIndex) {n val e = this[i]n val v = selector(e)n if (maxValue < v)
{n maxElem = e\n maxValue = v\n }n }n return maxElem}n/n/**n * Returns the first
element yielding the largest value of the given function or `null` if there are no elements.n * n * @sample
samples.collections.Collections.Aggregates.maxByOrNulln
*\/n@SinceKotlin("1.4")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun <R :
Comparable<R>> UByteArray.maxByOrNull(selector: (UByte) -> R): UByte? {n if (isEmpty()) return nulln
var maxElem = this[0]n val lastIndex = this.lastIndexn if (lastIndex == 0) return maxElemn var maxValue =
selector(maxElem)n for (i in 1..lastIndex) {n val e = this[i]n val v = selector(e)n if (maxValue < v)
{n maxElem = e\n maxValue = v\n }n }n return maxElem}n/n/**n * Returns the first
element yielding the largest value of the given function or `null` if there are no elements.n * n * @sample
samples.collections.Collections.Aggregates.maxByOrNulln
*\/n@SinceKotlin("1.4")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun <R :
Comparable<R>> UShortArray.maxByOrNull(selector: (UShort) -> R): UShort? {n if (isEmpty()) return nulln
var maxElem = this[0]n val lastIndex = this.lastIndexn if (lastIndex == 0) return maxElemn var maxValue =
selector(maxElem)n for (i in 1..lastIndex) {n val e = this[i]n val v = selector(e)n if (maxValue < v)
{n maxElem = e\n maxValue = v\n }n }n return maxElem}n/n/**n * Returns the largest
value among all values produced by [selector] functionn * applied to each element in the array.n * n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.n * n * @throws
NoSuchElementException if the array is empty.n
*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun

```



```

UIntArray.maxOf(selector: (UInt) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue =
            maxOf(maxValue, v)
    }
    return maxValue
}
/**
 * Returns the largest value among all values produced
 * by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector]
 * function is `NaN`, the returned result is `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.maxOf(selector: (ULong) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue =
            maxOf(maxValue, v)
    }
    return maxValue
}
/**
 * Returns the largest value among all values produced
 * by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector]
 * function is `NaN`, the returned result is `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.maxOf(selector: (UByte) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue =
            maxOf(maxValue, v)
    }
    return maxValue
}
/**
 * Returns the largest value among all values produced
 * by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector]
 * function is `NaN`, the returned result is `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.maxOf(selector: (UShort) -> Double): Double {
    if (isEmpty()) throw
        NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v =
            selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}
/**
 * Returns the
 * largest value among all values produced by [selector] function
 * applied to each element in the array.
 * If
 * any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 * @throws
 * NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.maxOf(selector: (UInt) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var
        maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue =
            maxOf(maxValue, v)
    }
    return maxValue
}
/**
 * Returns the largest value among all values produced
 * by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector]
 * function is `NaN`, the returned result is `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.maxOf(selector: (ULong) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var
        maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue =
            maxOf(maxValue, v)
    }
    return maxValue
}
/**
 * Returns the largest value among all values produced
 * by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector]
 * function is `NaN`, the returned result is `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.maxOf(selector: (UByte) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var
        maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue =
            maxOf(maxValue, v)
    }
    return maxValue
}
/**
 * Returns the largest value among all values produced
 * by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector]
 * function is `NaN`, the returned result is `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */

```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUShortArray.maxOf(selector: (UShort) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue =\n        maxOf(maxValue, v)\n    }\n    return maxValue\n}\n/**\n * Returns the largest value among all values produced\n * by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the\n * array is empty.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UIntArray.maxOf(selector: (UInt) -> R): R {\n    if (isEmpty()) throw\n    NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =\n        selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the\n * array.\n * @throws NoSuchElementException if the array is empty.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ULongArray.maxOf(selector: (ULong) -> R): R {\n    if (isEmpty()) throw\n    NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =\n        selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the\n * array.\n * @throws NoSuchElementException if the array is empty.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UByteArray.maxOf(selector: (UByte) -> R): R {\n    if (isEmpty()) throw\n    NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =\n        selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the\n * array.\n * @throws NoSuchElementException if the array is empty.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UShortArray.maxOf(selector: (UShort) -> R): R {\n    if (isEmpty()) throw\n    NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =\n        selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array\n * or `null` if there are no elements.\n * @throws NoSuchElementException if any of values produced by [selector] function is `NaN`, the returned\n * result is `NaN`.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUIntArray.maxOfOrNull(selector: (UInt) -> Double): Double? {\n    if (isEmpty()) return null\n    var maxValue =\n    selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * @throws NoSuchElementException if any of values produced by [selector]\n * function is `NaN`, the returned result is `NaN`.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nULongArray.maxOfOrNull(selector: (ULong) -> Double): Double? {\n    if (isEmpty()) return null\n    var\n    maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue =\n        maxOf(maxValue, v)\n    }\n    return maxValue\n}\n/**\n * Returns the largest value among all values produced
```

by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.maxOfOrNull(selector: (UByte) -> Double): Double? {\n if (isEmpty()) return null\n var maxValue
= selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue,
v)\n }\n return maxValue\n}\n\n**\n * Returns the largest value among all values produced by [selector]
function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced
by [selector] function is `NaN`, the returned result is `NaN`.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.maxOfOrNull(selector: (UShort) -> Double): Double? {\n if (isEmpty()) return null\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue =
maxOf(maxValue, v)\n }\n return maxValue\n}\n\n**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.maxOfOrNull(selector: (UInt) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n
}\n return maxValue\n}\n\n**\n * Returns the largest value among all values produced by [selector] function\n *
applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.maxOfOrNull(selector: (ULong) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n
}\n return maxValue\n}\n\n**\n * Returns the largest value among all values produced by [selector] function\n *
applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.maxOfOrNull(selector: (UByte) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n
}\n return maxValue\n}\n\n**\n * Returns the largest value among all values produced by [selector] function\n *
applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.maxOfOrNull(selector: (UShort) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n
}\n return maxValue\n}\n\n**\n * Returns the largest value among all values produced by [selector] function\n *
applied to each element in the array or `null` if there are no elements.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UIntArray.maxOfOrNull(selector: (UInt) -> R): R? {\n if (isEmpty()) return null\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n
maxValue = v\n }\n }\n return maxValue\n}\n\n**\n * Returns the largest value among all values

```

```

produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> ULongArray.maxOfOrNull(selector: (ULong) -> R): R? {\n if (isEmpty()) return null\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n
maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UByteArray.maxOfOrNull(selector: (UByte) -> R): R? {\n if (isEmpty()) return null\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n
maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UShortArray.maxOfOrNull(selector: (UShort) -> R): R? {\n if (isEmpty()) return null\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n
maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value according to the
provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n *
\n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.maxOfWith(comparator: Comparator<in R>, selector: (UInt) -> R): R {\n if (isEmpty()) throw
NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return
maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if
the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.maxOfWith(comparator: Comparator<in R>, selector: (ULong) -> R): R {\n if (isEmpty()) throw
NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return
maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if
the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.maxOfWith(comparator: Comparator<in R>, selector: (UByte) -> R): R {\n if (isEmpty()) throw
NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return
maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if
the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.maxOfWith(comparator: Comparator<in R>, selector: (UShort) -> R): R {\n if (isEmpty()) throw
NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =

```

```

selector(this[i])\n    if (comparator.compare(maxValue, v) < 0) {\n        maxValue = v\n    }\n }\n return
maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (UInt) -> R): R? {\n    if (isEmpty()) return
null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (ULong) -> R): R? {\n    if (isEmpty())
return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (UByte) -> R): R? {\n    if (isEmpty())
return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (UShort) -> R): R? {\n    if (isEmpty())
return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n *
Returns the largest element or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.maxOrNull(): UInt? {\n    if
(isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (max < e) max
= e\n    }\n    return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.maxOrNull(): ULong? {\n    if
(isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (max < e) max
= e\n    }\n    return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.maxOrNull(): UByte? {\n    if
(isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (max < e) max
= e\n    }\n    return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.maxOrNull(): UShort? {\n    if
(isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (max < e) max
= e\n    }\n    return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided
[comparator].\n * *\n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
UIntArray.maxWith(comparator: Comparator<in UInt>): UInt {\n    if (isEmpty()) throw
NoSuchElementException()\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if
(comparator.compare(max, e) < 0) max = e\n    }\n    return max\n}\n\n/**\n * Returns the first element having the

```

```

largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is
empty.\n *\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
ULongArray.maxWith(comparator: Comparator<in ULong>): ULong {\n if (isEmpty()) throw
NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is
empty.\n *\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
UByteArray.maxWith(comparator: Comparator<in UByte>): UByte {\n if (isEmpty()) throw
NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is
empty.\n *\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("maxWithOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
UShortArray.maxWith(comparator: Comparator<in UShort>): UShort {\n if (isEmpty()) throw
NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.maxWithOrNull(comparator:
Comparator<in UInt>): UInt? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n
val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the
first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.maxWithOrNull(comparator:
Comparator<in ULong>): ULong? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n
val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the
first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.maxWithOrNull(comparator:
Comparator<in UByte>): UByte? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n
val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the
first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.maxWithOrNull(comparator:
Comparator<in UShort>): UShort? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n
val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the
smallest element.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
UIntArray.min(): UInt {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in
1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the smallest
element.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
ULongArray.min(): ULong {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in
1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the smallest
element.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
UByteArray.min(): UByte {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in

```


element yielding the smallest value of the given function or `null` if there are no elements.

`samples.collections.Collections.Aggregates.minByOrNull`

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ULongArray.minByOrNull(selector: (ULong) -> R): ULong? {\n    if (isEmpty()) return null\n    var minElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return minElem\n    var minValue = selector(minElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (minValue > v)\n            minElem = e\n            minValue = v\n    }\n    return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
```

`samples.collections.Collections.Aggregates.minByOrNull`

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UByteArray.minByOrNull(selector: (UByte) -> R): UByte? {\n    if (isEmpty()) return null\n    var minElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return minElem\n    var minValue = selector(minElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (minValue > v)\n            minElem = e\n            minValue = v\n    }\n    return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
```

`samples.collections.Collections.Aggregates.minByOrNull`

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UShortArray.minByOrNull(selector: (UShort) -> R): UShort? {\n    if (isEmpty()) return null\n    var minElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return minElem\n    var minValue = selector(minElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (minValue > v)\n            minElem = e\n            minValue = v\n    }\n    return minElem\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws
```

`NoSuchElementException` if the array is empty.

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.minOf(selector: (UInt) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.minOf(selector: (ULong) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.minOf(selector: (UByte) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.minOf(selector: (UShort) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n
```



```

    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue =
minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]
function is `NaN`, the\n returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution
ByLambdaReturnType\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun
UIntArray.minOf(selector: (UInt) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var
minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue =
minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution
ByLambdaReturnType\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun
UIntArray.minOf(selector: (UInt) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var
minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue =
minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution
ByLambdaReturnType\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun
UByteArray.minOf(selector: (UByte) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var
minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue =
minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution
ByLambdaReturnType\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun
UShortArray.minOf(selector: (UShort) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var
minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue =
minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the
array is empty.\n
*\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution
ByLambdaReturnType\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun <R :
Comparable<R>> UIntArray.minOf(selector: (UInt) -> R): R {\n    if (isEmpty()) throw
NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =
selector(this[i])\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the
array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution
ByLambdaReturnType\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun <R :
Comparable<R>> ULongArray.minOf(selector: (ULong) -> R): R {\n    if (isEmpty()) throw
NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =
selector(this[i])\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the
array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution
ByLambdaReturnType\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun <R :

```

```

Comparable<R>> UByteArray.minOf(selector: (UByte) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

Returns the smallest value among all values produced by [selector] function applied to each element in the array.
@throws NoSuchElementException if the array is empty.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UShortArray.minOf(selector: (UShort) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

Returns the smallest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.minOfOrNull(selector: (UInt) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

Returns the smallest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.minOfOrNull(selector: (ULong) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

Returns the smallest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.minOfOrNull(selector: (UByte) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

Returns the smallest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.minOfOrNull(selector: (UShort) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

Returns the smallest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.minOfOrNull(selector: (UInt) -> Float): Float? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

Returns the smallest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.minOfOrNull(selector: (ULong) -> Float): Float? {\n if (isEmpty()) return null\n var minValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n
}\n return minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n *
applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.minOfOrNull(selector: (UByte) -> Float): Float? {\n if (isEmpty()) return null\n var minValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n
}\n return minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n *
applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.minOfOrNull(selector: (UShort) -> Float): Float? {\n if (isEmpty()) return null\n var minValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n
}\n return minValue}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n *
applied to each element in the array or `null` if there are no elements.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UIntArray.minOfOrNull(selector: (UInt) -> R): R? {\n if (isEmpty()) return null\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n
minValue = v\n }\n }\n return minValue}\n\n/**\n * Returns the smallest value among all values
produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> ULongArray.minOfOrNull(selector: (ULong) -> R): R? {\n if (isEmpty()) return null\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n
minValue = v\n }\n }\n return minValue}\n\n/**\n * Returns the smallest value among all values
produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UByteArray.minOfOrNull(selector: (UByte) -> R): R? {\n if (isEmpty()) return null\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n
minValue = v\n }\n }\n return minValue}\n\n/**\n * Returns the smallest value among all values
produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UShortArray.minOfOrNull(selector: (UShort) -> R): R? {\n if (isEmpty()) return null\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n
minValue = v\n }\n }\n return minValue}\n\n/**\n * Returns the smallest value according to the
provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n *
\n * @throws NoSuchElementException if the array is empty.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.minOfWith(comparator: Comparator<in R>, selector: (UInt) -> R): R {\n if (isEmpty()) throw

```

```

NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =
selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return
minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if
the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.minOfWith(comparator: Comparator<in R>, selector: (ULong) -> R): R {\n    if (isEmpty()) throw
NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =
selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return
minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if
the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.minOfWith(comparator: Comparator<in R>, selector: (UByte) -> R): R {\n    if (isEmpty()) throw
NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =
selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return
minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.minOfWith(comparator: Comparator<in R>, selector: (UShort) -> R): R {\n    if (isEmpty()) throw
NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =
selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return
minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (UInt) -> R): R? {\n    if (isEmpty()) return
null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (ULong) -> R): R? {\n    if (isEmpty())
return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (UByte) -> R): R? {\n    if (isEmpty())
return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n

```

```

function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (UShort) -> R): R? {\n if (isEmpty())
return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue, v) > 0) {\n minValue = v\n } }\n return minValue}\n\n/**\n *
Returns the smallest element or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.minOrNull(): UInt? {\n if
(isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min =
e\n }\n return min}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.minOrNull(): ULong? {\n if
(isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min =
e\n }\n return min}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.minOrNull(): UByte? {\n if
(isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min =
e\n }\n return min}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.minOrNull(): UShort? {\n if
(isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min =
e\n }\n return min}\n\n/**\n * Returns the first element having the smallest value according to the provided
[comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
UIntArray.minWith(comparator: Comparator<in UInt>): UInt {\n if (isEmpty()) throw
NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(min, e) > 0) min = e\n }\n return min}\n\n/**\n * Returns the first element having the
smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is
empty.\n * \n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
ULongArray.minWith(comparator: Comparator<in ULong>): ULong {\n if (isEmpty()) throw
NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(min, e) > 0) min = e\n }\n return min}\n\n/**\n * Returns the first element having the
smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is
empty.\n * \n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
UByteArray.minWith(comparator: Comparator<in UByte>): UByte {\n if (isEmpty()) throw
NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(min, e) > 0) min = e\n }\n return min}\n\n/**\n * Returns the first element having the
smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is
empty.\n * \n@SinceKotlin("1.7")\n@kotlin.jvm.JvmName("minWithOrThrow-
U")\n@ExperimentalUnsignedTypes\n@Suppress("CONFLICTING_OVERLOADS")\npublic fun
UShortArray.minWith(comparator: Comparator<in UShort>): UShort {\n if (isEmpty()) throw
NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if
(comparator.compare(min, e) > 0) min = e\n }\n return min}\n\n/**\n * Returns the first element having the
smallest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.minWithOrNull(comparator:
Comparator<in UInt>): UInt? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n
val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min}\n\n/**\n * Returns the first
element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n

```

```

*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.minWithOrNull(comparator:
Comparator<in ULong>): ULong? {\n    if (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n
        val e = this[i]\n        if (comparator.compare(min, e) > 0) min = e\n    }\n    return min\n}\n\n/**\n * Returns the
first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.minWithOrNull(comparator:
Comparator<in UByte>): UByte? {\n    if (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n
        val e = this[i]\n        if (comparator.compare(min, e) > 0) min = e\n    }\n    return min\n}\n\n/**\n * Returns the
first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.minWithOrNull(comparator:
Comparator<in UShort>): UShort? {\n    if (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n
        val e = this[i]\n        if (comparator.compare(min, e) > 0) min = e\n    }\n    return min\n}\n\n/**\n * Returns
`true` if the array has no elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.none(): Boolean {\n    return isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n *
@sample samples.collections.Collections.Aggregates.none\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.none(): Boolean {\n    return isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n *
@sample samples.collections.Collections.Aggregates.none\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.none(): Boolean {\n    return isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n *
@sample samples.collections.Collections.Aggregates.none\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.none(): Boolean {\n    return isEmpty()\n}\n\n/**\n * Returns `true` if no elements match the given
[predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.none(predicate: (UInt) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return
false\n    return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.none(predicate: (ULong) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return
false\n    return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.none(predicate: (UByte) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return
false\n    return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.none(predicate: (UShort) -> Boolean): Boolean {\n    for (element in this) if (predicate(element))
return false\n    return true\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself
afterwards.\n * \n *\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UIntArray.onEach(action: (UInt) -> Unit): UIntArray {\n    return apply { for (element in this)
action(element) }\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself
afterwards.\n * \n *\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.onEach(action: (ULong) -> Unit): ULongArray {\n    return apply { for (element in this)
action(element) }\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself
afterwards.\n * \n *\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UByteArray.onEach(action: (UByte) -> Unit): UByteArray {\n    return apply { for (element in this)
action(element) }\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself

```

```

afterwards.\n *^@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UShortArray.onEach(action: (UShort) -> Unit): UShortArray {\n    return apply { for (element in this)
action(element) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the
element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element
and the element itself\n * and performs the action on the element.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.onEachIndexed(action: (index: Int, UInt) -> Unit): UIntArray {\n    return apply {
forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index
with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an
element and the element itself\n * and performs the action on the element.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.onEachIndexed(action: (index: Int, ULong) -> Unit): ULongArray {\n    return apply {
forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index
with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an
element and the element itself\n * and performs the action on the element.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.onEachIndexed(action: (index: Int, UByte) -> Unit): UByteArray {\n    return apply {
forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index
with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an
element and the element itself\n * and performs the action on the element.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.onEachIndexed(action: (index: Int, UShort) -> Unit): UShortArray {\n    return apply {
forEachIndexed(action) }\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation]
from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is
empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its
receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n *
and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.reduce(operation: (acc: UInt, UInt) -> UInt): UInt {\n    if (isEmpty())\n        throw
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in
1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current
accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty
in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n *
@param [operation] function that takes current accumulator value and an element,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.reduce(operation: (acc: ULong, ULong) -> ULong): ULong {\n    if (isEmpty())\n        throw
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in
1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current
accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty
in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n *
@param [operation] function that takes current accumulator value and an element,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.reduce(operation: (acc: UByte, UByte) -> UByte): UByte {\n    if (isEmpty())\n        throw
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in

```


throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample

samples.collections.Collections.Aggregates.reduceOrNull\n

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUIntArray.reduceIndexedOrNull(operation: (index: Int, acc: UInt, UInt) -> UInt): UInt? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nULongArray.reduceIndexedOrNull(operation: (index: Int, acc: ULong, ULong) -> ULong): ULong? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUByteArray.reduceIndexedOrNull(operation: (index: Int, acc: UByte, UByte) -> UByte): UByte? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUShortArray.reduceIndexedOrNull(operation: (index: Int, acc: UShort, UShort) -> UShort): UShort? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reduceOrNull(operation: (acc: UInt, UInt) -> UInt): UInt? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
```

```

@kotlin.internal.InlineOnly\npublic inline fun ULongArray.reduceOrNull(operation: (acc: ULong, ULong) ->
ULong): ULong? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex)
{\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates
value starting with the first element and applying [operation] from left to right\n * to current accumulator value and
each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current
accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UByteArray.reduceOrNull(operation: (acc: UByte, UByte) ->
UByte): UByte? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex)
{\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates
value starting with the first element and applying [operation] from left to right\n * to current accumulator value and
each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current
accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UShortArray.reduceOrNull(operation: (acc: UShort, UShort) ->
UShort): UShort? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex)
{\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates
value starting with the last element and applying [operation] from right to left\n * to each element and current
accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected
way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param
[operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n
*/\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.reduceRight(operation: (UInt, acc: UInt) -> UInt): UInt {\n var index = lastIndex\n if (index < 0)
throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = get(index--)\n
while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return
accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the
array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its
receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n *
and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n
*/\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.reduceRight(operation: (ULong, acc: ULong) -> ULong): ULong {\n var index = lastIndex\n if
(index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator =
get(index--)\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return
accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the
array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its
receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n *
and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n
*/\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.reduceRight(operation: (UByte, acc: UByte) -> UByte): UByte {\n var index = lastIndex\n if (index
< 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = get(index--)\n
while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return
accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the
array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its
receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n *
and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n
*/

```

accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample

samples.collections.Collections.Aggregates.reduceRight\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUShortArray.reduceRight(operation: (UShort, acc: UShort) -> UShort): UShort {\n    var index = lastIndex\n    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator =\n    get(index--)\n    while (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return\n    accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use
```

[reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUIntArray.reduceRightIndexed(operation: (index: Int, UInt, acc: UInt) -> UInt): UInt {\n    var index = lastIndex\n    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator =\n    get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n        --index\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use
```

[reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nULongArray.reduceRightIndexed(operation: (index: Int, ULong, acc: ULong) -> ULong): ULong {\n    var index =\n    lastIndex\n    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n    var\n    accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index),\n        accumulator)\n        --index\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use
```

[reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUByteArray.reduceRightIndexed(operation: (index: Int, UByte, acc: UByte) -> UByte): UByte {\n    var index =\n    lastIndex\n    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n    var\n    accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index),\n        accumulator)\n        --index\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
```

samples.collections.Collections.Aggregates.reduceRight\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUByteArray.reduceRightIndexed(operation: (index: Int, UByte, acc: UByte) -> UByte): UByte {\n    var index =\n    lastIndex\n    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n    var\n    accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index),\n        accumulator)\n        --index\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
```

samples.collections.Collections.Aggregates.reduceRight\n

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.reduceRightIndexed(operation: (index: Int, UShort, acc: UShort) -> UShort): UShort {\n  var index =
lastIndex\n  if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n  var
accumulator = get(index--)\n  while (index >= 0) {\n    accumulator = operation(index, get(index),
accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last
element and applying [operation] from right to left\n * to each element with its index in the original array and
current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that
takes the index of an element, the element itself and current accumulator value,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.reduceRightIndexedOrNull(operation: (index: Int, UInt, acc: UInt) -> UInt): UInt? {\n  var index =
lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n *
Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with
its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator value,\n
* and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.reduceRightIndexedOrNull(operation: (index: Int, ULong, acc: ULong) -> ULong): ULong? {\n  var
index = lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n *
Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with
its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator value,\n
* and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.reduceRightIndexedOrNull(operation: (index: Int, UByte, acc: UByte) -> UByte): UByte? {\n  var
index = lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n *
Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with
its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator value,\n
* and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.reduceRightIndexedOrNull(operation: (index: Int, UShort, acc: UShort) -> UShort): UShort? {\n  var
index = lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n *
Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and
current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that
takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reduceRightOrNull(operation: (UInt, acc: UInt) -> UInt):
UInt? {\n  var index = lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index
>= 0) {\n    accumulator = operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n *

```

Accumulates value starting with the last element and applying [operation] from right to left to each element and current accumulator value. Returns `null` if the array is empty. @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value. @sample samples.collections.Collections.Aggregates.reduceRightOrNull

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.reduceRightOrNull(operation: (ULong, acc: ULong) -> ULong): ULong? {\n    var index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return accumulator\n}
```

Accumulates value starting with the last element and applying [operation] from right to left to each element and current accumulator value. Returns `null` if the array is empty. @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value. @sample samples.collections.Collections.Aggregates.reduceRightOrNull

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.reduceRightOrNull(operation: (UByte, acc: UByte) -> UByte): UByte? {\n    var index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return accumulator\n}
```

Accumulates value starting with the last element and applying [operation] from right to left to each element and current accumulator value. Returns `null` if the array is empty. @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value. @sample samples.collections.Collections.Aggregates.reduceRightOrNull

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.reduceRightOrNull(operation: (UShort, acc: UShort) -> UShort): UShort? {\n    var index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return accumulator\n}
```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list. @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value. @sample samples.collections.Collections.Aggregates.runningFold

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nUIntArray.runningFold(initial: R, operation: (acc: R, UInt) -> R): List<R> {\n    if (isEmpty()) return listOf(initial)\n    val result = ArrayList<R>(size + 1).apply { add(initial) }\n    var accumulator = initial\n    for (element in this) {\n        accumulator = operation(accumulator, element)\n        result.add(accumulator)\n    }\n    return result\n}
```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list. @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value. @sample samples.collections.Collections.Aggregates.runningFold

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nULongArray.runningFold(initial: R, operation: (acc: R, ULong) -> R): List<R> {\n    if (isEmpty()) return listOf(initial)\n    val result = ArrayList<R>(size + 1).apply { add(initial) }\n    var accumulator = initial\n    for (element in this) {\n        accumulator = operation(accumulator, element)\n        result.add(accumulator)\n    }\n    return result\n}
```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list. @param [operation] function that takes current accumulator value and an

element, and calculates the next accumulator value.

```

\n * \n * @sample
samples.collections.Collections.Aggregates.runningFold

```

```

*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.runningFold(initial: R, operation: (acc: R, UByte) -> R): List<R> {\n if (isEmpty()) return
listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for
(element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n
return result\n}\n\n**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n *
\n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the
previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an
element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningFold

```

```

*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.runningFold(initial: R, operation: (acc: R, UShort) -> R): List<R> {\n if (isEmpty()) return
listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for
(element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n
return result\n}\n\n**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element, its index in the original array and current accumulator value that
starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the
index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold

```

```

*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, UInt) -> R): List<R> {\n if (isEmpty())
return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n
for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n }\n return result\n}\n\n**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function
should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold

```

```

*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, ULong) -> R): List<R> {\n if
(isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator =
initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n }\n return result\n}\n\n**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function
should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold

```

```

*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, UByte) -> R): List<R> {\n if
(isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator =
initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n }\n return result\n}\n\n**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and

```

current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold

```

*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, UShort) -> R): List<R> {\n  if
(isEmpty()) return listOf(initial)\n  val result = ArrayList<R>(size + 1).apply { add(initial) }\n  var accumulator =
initial\n  for (index in indices) {\n    accumulator = operation(index, accumulator, this[index])\n    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element and current accumulator value that
starts with the first element of this array.\n * \n * Note that `acc` value passed to [operation] function should not be
mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that
takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.runningReduce(operation: (acc: UInt, UInt) -> UInt): List<UInt> {\n  if (isEmpty()) return
emptyList()\n  var accumulator = this[0]\n  val result = ArrayList<UInt>(size).apply { add(accumulator) }\n  for
(index in 1 until size) {\n    accumulator = operation(accumulator, this[index])\n    result.add(accumulator)\n  }\n
return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element and current accumulator value that starts with the first element of
this array.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would
affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value
and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.runningReduce(operation: (acc: ULong, ULong) -> ULong): List<ULong> {\n  if (isEmpty()) return
emptyList()\n  var accumulator = this[0]\n  val result = ArrayList<ULong>(size).apply { add(accumulator) }\n
for (index in 1 until size) {\n    accumulator = operation(accumulator, this[index])\n    result.add(accumulator)\n
}\n  return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element and current accumulator value that starts with the first element of
this array.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would
affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value
and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.runningReduce(operation: (acc: UByte, UByte) -> UByte): List<UByte> {\n  if (isEmpty()) return
emptyList()\n  var accumulator = this[0]\n  val result = ArrayList<UByte>(size).apply { add(accumulator) }\n
for (index in 1 until size) {\n    accumulator = operation(accumulator, this[index])\n    result.add(accumulator)\n
}\n  return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element and current accumulator value that starts with the first element of
this array.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would
affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value
and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.runningReduce(operation: (acc: UShort, UShort) -> UShort): List<UShort> {\n  if (isEmpty()) return
emptyList()\n  var accumulator = this[0]\n  val result = ArrayList<UShort>(size).apply { add(accumulator) }\n
for (index in 1 until size) {\n    accumulator = operation(accumulator, this[index])\n    result.add(accumulator)\n
}

```

```

    }
    return result
}

/** Returns a list containing successive accumulation values generated by applying
 [operation] from left to right to each element, its index in the original array and current accumulator value that
 starts with the first element of this array.
 * Note that `acc` value passed to [operation] function should not be mutated;
 otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes the index of an element, current accumulator value
 and the element itself, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningReduce
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.runningReduceIndexed(operation: (index: Int, acc: UInt, UInt) -> UInt): List<UInt> {
    if (isEmpty())
        return emptyList()
    var accumulator = this[0]
    val result = ArrayList<UInt>(size).apply {
        add(accumulator)
    }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/** Returns a list containing successive accumulation
 values generated by applying [operation] from left to right to each element, its index in the original array and
 current accumulator value that starts with the first element of this array.
 * Note that `acc` value passed to
 [operation] function should not be mutated;
 otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes the index of an element, current accumulator value
 and the element
 itself, and calculates the next accumulator value.
 * @sample
 samples.collections.Collections.Aggregates.runningReduce
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.runningReduceIndexed(operation: (index: Int, acc: ULong, ULong) -> ULong): List<ULong> {
    if (isEmpty())
        return emptyList()
    var accumulator = this[0]
    val result = ArrayList<ULong>(size).apply {
        add(accumulator)
    }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/** Returns a list containing successive accumulation
 values generated by applying [operation] from left to right to each element, its index in the original array and
 current accumulator value that starts with the first element of this array.
 * Note that `acc` value passed to
 [operation] function should not be mutated;
 otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes the index of an element, current accumulator value
 and the element
 itself, and calculates the next accumulator value.
 * @sample
 samples.collections.Collections.Aggregates.runningReduce
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.runningReduceIndexed(operation: (index: Int, acc: UByte, UByte) -> UByte): List<UByte> {
    if (isEmpty())
        return emptyList()
    var accumulator = this[0]
    val result = ArrayList<UByte>(size).apply {
        add(accumulator)
    }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/** Returns a list containing successive accumulation
 values generated by applying [operation] from left to right to each element, its index in the original array and
 current accumulator value that starts with the first element of this array.
 * Note that `acc` value passed to
 [operation] function should not be mutated;
 otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes the index of an element, current accumulator value
 and the element
 itself, and calculates the next accumulator value.
 * @sample
 samples.collections.Collections.Aggregates.runningReduce
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.runningReduceIndexed(operation: (index: Int, acc: UShort, UShort) -> UShort): List<UShort> {
    if (isEmpty())
        return emptyList()
    var accumulator = this[0]
    val result = ArrayList<UShort>(size).apply {
        add(accumulator)
    }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/** Returns a list containing successive accumulation
 values generated by applying [operation] from left to right to each element and current accumulator value that
 starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated;
 otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes current
 accumulator value and an element, and calculates the next accumulator value.
 * @sample

```



```

samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun <R> UIntArray.scan(initial: R, operation: (acc: R, UInt) -> R):
List<R> {\n    return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element and current accumulator value that
starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current
accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.scan(initial: R, operation: (acc: R, ULong) -> R):
List<R> {\n    return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element and current accumulator value that
starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current
accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.scan(initial: R, operation: (acc: R, UByte) -> R):
List<R> {\n    return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element and current accumulator value that
starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current
accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun <R> UShortArray.scan(initial: R, operation: (acc: R, UShort) -> R):
List<R> {\n    return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function
should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun <R> UIntArray.scanIndexed(initial: R, operation: (index: Int, acc: R,
UInt) -> R): List<R> {\n    return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing
successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in
the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed
to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n *
\n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element
itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.scanIndexed(initial: R, operation: (index: Int, acc:
R, ULong) -> R): List<R> {\n    return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element,
its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc`
value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in

```



```

UByteArray.sumByDouble(selector: (UByte) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector] function applied to each element in the array.
@Deprecated("Use sumOf instead.", ReplaceWith("this.sumOf(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.5")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.sumByDouble(selector: (UShort) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector] function applied to each element in the array.
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfDouble")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.sumOf(selector: (UInt) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector] function applied to each element in the array.
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfDouble")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.sumOf(selector: (ULong) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector] function applied to each element in the array.
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfDouble")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.sumOf(selector: (UByte) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector] function applied to each element in the array.
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfDouble")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.sumOf(selector: (UShort) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector] function applied to each element in the array.
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfInt")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.sumOf(selector: (UInt) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector] function applied to each element in the array.
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfInt")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.sumOf(selector: (ULong) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector] function applied to each element in the array.
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfInt")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.sumOf(selector: (UByte) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

```

```

return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in
the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfInt")\n
@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.sumOf(selector:
(UShort) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n
return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in
the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfLong")
\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.sumOf(selector: (UInt)
-> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this) {\n        sum += selector(element)\n    }\n
return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in
the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfLong")
\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.sumOf(selector:
(ULong) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this) {\n        sum +=
selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfLong")
\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.sumOf(selector:
(UByte) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this) {\n        sum +=
selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfLong")
\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.sumOf(selector:
(UShort) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this) {\n        sum +=
selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfUInt")\n
\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inline
Only\npublic inline fun UIntArray.sumOf(selector: (UInt) -> UInt): UInt {\n    var sum: UInt = 0.toUInt()\n    for
(element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values
produced by [selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfUInt")\n
\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inline
Only\npublic inline fun ULongArray.sumOf(selector: (ULong) -> UInt): UInt {\n    var sum: UInt = 0.toUInt()\n
for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all
values produced by [selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfUInt")\n
\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inline
Only\npublic inline fun UByteArray.sumOf(selector: (UByte) -> UInt): UInt {\n    var sum: UInt = 0.toUInt()\n

```

```

for (element in this) {
    sum += selector(element)
}
return sum
}

Returns the sum of all values produced by [selector] function applied to each element in the array.

@SinceKotlin("1.5")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfUInt")
@ExperimentalUnsignedTypes
@WasExperimental(ExperimentalUnsignedTypes::class)
@kotlin.internal.InlineOnly
public inline fun UShortArray.sumOf(selector: (UShort) -> UInt): UInt {
    var sum: UInt = 0.toUInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

Returns the sum of all values produced by [selector] function applied to each element in the array.

@SinceKotlin("1.5")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfULong")
@ExperimentalUnsignedTypes
@WasExperimental(ExperimentalUnsignedTypes::class)
@kotlin.internal.InlineOnly
public inline fun UIntArray.sumOf(selector: (UInt) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

Returns the sum of all values produced by [selector] function applied to each element in the array.

@SinceKotlin("1.5")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfULong")
@ExperimentalUnsignedTypes
@WasExperimental(ExperimentalUnsignedTypes::class)
@kotlin.internal.InlineOnly
public inline fun ULongArray.sumOf(selector: (ULong) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

Returns the sum of all values produced by [selector] function applied to each element in the array.

@SinceKotlin("1.5")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfULong")
@ExperimentalUnsignedTypes
@WasExperimental(ExperimentalUnsignedTypes::class)
@kotlin.internal.InlineOnly
public inline fun UByteArray.sumOf(selector: (UByte) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

Returns the sum of all values produced by [selector] function applied to each element in the array.

@SinceKotlin("1.5")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfULong")
@ExperimentalUnsignedTypes
@WasExperimental(ExperimentalUnsignedTypes::class)
@kotlin.internal.InlineOnly
public inline fun UShortArray.sumOf(selector: (UShort) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
* The returned list has length of the shortest collection.
* @sample samples.collections.Iterables.Operations.zipIterable

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun <R> UIntArray.zip(other: Array<out R>): List<Pair<UInt, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
* The returned list has length of the shortest collection.
* @sample samples.collections.Iterables.Operations.zipIterable

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun <R> ULongArray.zip(other: Array<out R>): List<Pair<ULong, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
* The returned list has length of the shortest collection.
* @sample samples.collections.Iterables.Operations.zipIterable

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun <R> UByteArray.zip(other: Array<out R>): List<Pair<UByte, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
* The returned list has length of the shortest collection.
* @sample samples.collections.Iterables.Operations.zipIterable

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun <R> UShortArray.zip(other: Array<out R>): List<Pair<UShort, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

Returns a list of values built from the elements of `this` array and the [other] array with the same index
* using the provided

```

```

[transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n *
\n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
UIntArray.zip(other: Array<out R>, transform: (a: UInt, b: R) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
ULongArray.zip(other: Array<out R>, transform: (a: ULong, b: R) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
UByteArray.zip(other: Array<out R>, transform: (a: UByte, b: R) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
UShortArray.zip(other: Array<out R>, transform: (a: UShort, b: R) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with
the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UIntArray.zip(other:
Iterable<R>): List<Pair<UInt, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs
built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of
the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> ULongArray.zip(other:
Iterable<R>): List<Pair<ULong, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs
built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of
the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UByteArray.zip(other:
Iterable<R>): List<Pair<UByte, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs
built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of
the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UShortArray.zip(other:
Iterable<R>): List<Pair<UShort, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of
values built from the elements of `this` array and the [other] collection with the same index\n * using the provided
[transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n *
\n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
UIntArray.zip(other: Iterable<R>, transform: (a: UInt, b: R) -> V): List<V> {\n    val arraySize = size\n    val list =

```

```

ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n  var i = 0\n  for (element in other) {\n
if (i >= arraySize) break\n    list.add(transform(this[i++], element))\n  }\n  return list\n}\n\n/**\n * Returns a
list of values built from the elements of `this` array and the [other] collection with the same index\n * using the
provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
ULongArray.zip(other: Iterable<R>, transform: (a: ULong, b: R) -> V): List<V> {\n  val arraySize = size\n  val
list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n  var i = 0\n  for (element in other)
{\n    if (i >= arraySize) break\n    list.add(transform(this[i++], element))\n  }\n  return list\n}\n\n/**\n *
Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n *
using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the
shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
UByteArray.zip(other: Iterable<R>, transform: (a: UByte, b: R) -> V): List<V> {\n  val arraySize = size\n  val
list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n  var i = 0\n  for (element in other) {\n
if (i >= arraySize) break\n    list.add(transform(this[i++], element))\n  }\n  return list\n}\n\n/**\n * Returns a
list of values built from the elements of `this` array and the [other] collection with the same index\n * using the
provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
UShortArray.zip(other: Iterable<R>, transform: (a: UShort, b: R) -> V): List<V> {\n  val arraySize = size\n  val
list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n  var i = 0\n  for (element in other)
{\n    if (i >= arraySize) break\n    list.add(transform(this[i++], element))\n  }\n  return list\n}\n\n/**\n *
Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The
returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun UIntArray.zip(other: UIntArray):
List<Pair<UInt, UInt>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from
the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the
shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun ULongArray.zip(other: ULongArray):
List<Pair<ULong, ULong>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built
from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the
shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun UByteArray.zip(other: UByteArray):
List<Pair<UByte, UByte>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built
from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the
shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun UShortArray.zip(other: UShortArray):
List<Pair<UShort, UShort>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of values built
from the elements of `this` array and the [other] array with the same index\n * using the provided [transform]
function applied to each pair of elements.\n * The returned list has length of the shortest array.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V>
UIntArray.zip(other: UIntArray, transform: (a: UInt, b: UInt) -> V): List<V> {\n  val size = minOf(size,
other.size)\n  val list = ArrayList<V>(size)\n  for (i in 0 until size) {\n    list.add(transform(this[i], other[i]))\n
}\n  return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned

```

```

list has length of the shortest array.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V>
ULongArray.zip(other: ULongArray, transform: (a: ULong, b: ULong) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest array.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V>
UByteArray.zip(other: UByteArray, transform: (a: UByte, b: UByte) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest array.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V>
UShortArray.zip(other: UShortArray, transform: (a: UShort, b: UShort) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*\n@kotlin.jvm.JvmName("sumOfUInt")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedT
ypes::class)\npublic fun Array<out UInt>.sum(): UInt {\n    var sum: UInt = 0u\n    for (element in this) {\n        sum
+= element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*\n@kotlin.jvm.JvmName("sumOfULong")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Array<out ULong>.sum(): ULong {\n    var sum: ULong = 0uL\n    for (element in this)
{\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*\n@kotlin.jvm.JvmName("sumOfUByte")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Array<out UByte>.sum(): UInt {\n    var sum: UInt = 0u\n    for (element in this) {\n
sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*\n@kotlin.jvm.JvmName("sumOfUShort")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Array<out UShort>.sum(): UInt {\n    var sum: UInt = 0u\n    for (element in this) {\n
sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.sum(): UInt {\n    return storage.sum().toUInt()\n}\n\n/**\n * Returns the sum of all elements in the
array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.sum(): ULong {\n    return storage.sum().toULong()\n}\n\n/**\n * Returns the sum of all elements in
the array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun UByteArray.sum(): UInt {\n    return sumOf { it.toUInt() }\n}\n\n/**\n * Returns the sum of all elements in the
array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.sum(): UInt {\n    return sumOf { it.toUInt() }\n}\n\n", "*/\n * Copyright 2010-2022 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("UCollectionsKt")\n\npackage
kotlin.collections\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n/\n\nimport kotlin.random.*\nimport
kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns an array of UByte containing all of the
elements of this collection.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
Collection<UByte>.toUByteArray(): UByteArray {\n    val result = UByteArray(size)\n    var index = 0\n    for

```



```

* Returns the greater of the given values.\n *^@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
maxOf(a: UInt, vararg other: UInt): UInt {\n    var max = a\n    for (e in other) max = maxOf(max, e)\n    return
max}\n}\n\n/**\n * Returns the greater of the given values.\n
*^@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun maxOf(a: ULong, vararg other: ULong):
ULong {\n    var max = a\n    for (e in other) max = maxOf(max, e)\n    return max}\n}\n\n/**\n * Returns the greater
of the given values.\n *^@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun maxOf(a: UByte,
vararg other: UByte): UByte {\n    var max = a\n    for (e in other) max = maxOf(max, e)\n    return max}\n}\n\n/**
* Returns the greater of the given values.\n *^@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
maxOf(a: UShort, vararg other: UShort): UShort {\n    var max = a\n    for (e in other) max = maxOf(max, e)\n
return max}\n}\n\n/**\n * Returns the smaller of two values.\n
*^@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun minOf(a: UInt, b:
UInt): UInt {\n    return if (a <= b) a else b}\n}\n\n/**\n * Returns the smaller of two values.\n
*^@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun minOf(a: ULong,
b: ULong): ULong {\n    return if (a <= b) a else b}\n}\n\n/**\n * Returns the smaller of two values.\n
*^@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun minOf(a: UByte,
b: UByte): UByte {\n    return if (a <= b) a else b}\n}\n\n/**\n * Returns the smaller of two values.\n
*^@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun minOf(a: UShort,
b: UShort): UShort {\n    return if (a <= b) a else b}\n}\n\n/**\n * Returns the smaller of three values.\n
*^@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline fun minOf(a: UInt, b: UInt, c: UInt): UInt {\n    return minOf(a, minOf(b, c))}\n}\n\n/**\n * Returns
the smaller of three values.\n
*^@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline fun minOf(a: ULong, b: ULong, c: ULong): ULong {\n    return minOf(a, minOf(b, c))}\n}\n\n/**\n *
Returns the smaller of three values.\n
*^@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline fun minOf(a: UByte, b: UByte, c: UByte): UByte {\n    return minOf(a, minOf(b, c))}\n}\n\n/**\n *
Returns the smaller of three values.\n
*^@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline fun minOf(a: UShort, b: UShort, c: UShort): UShort {\n    return minOf(a, minOf(b, c))}\n}\n\n/**\n *
Returns the smaller of the given values.\n *^@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
minOf(a: UInt, vararg other: UInt): UInt {\n    var min = a\n    for (e in other) min = minOf(min, e)\n    return
min}\n}\n\n/**\n * Returns the smaller of the given values.\n
*^@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun minOf(a: ULong, vararg other: ULong):
ULong {\n    var min = a\n    for (e in other) min = minOf(min, e)\n    return min}\n}\n\n/**\n * Returns the
smaller of the given values.\n *^@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun minOf(a: UByte,
vararg other: UByte): UByte {\n    var min = a\n    for (e in other) min = minOf(min, e)\n    return min}\n}\n\n/**
Returns the smaller of the given values.\n *^@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
minOf(a: UShort, vararg other: UShort): UShort {\n    var min = a\n    for (e in other) min = minOf(min, e)\n
return min}\n}\n\n"/**\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n *
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*^@n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("URangesKt")\n\npackage
kotlin.ranges\n\n/\n\nNOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n\n See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n\nimport kotlin.random.*\n\n/**\n * Returns the
first element.\n * \n * @throws NoSuchElementException if the progression is empty.\n
*^@SinceKotlin("1.7")\npublic fun UIntProgression.first(): UInt {\n    if (isEmpty())\n        throw
NoSuchElementException("Progression $this is empty.")\n    return this.first}\n}\n\n/**\n * Returns the first
element.\n * \n * @throws NoSuchElementException if the progression is empty.\n
*^@SinceKotlin("1.7")\npublic fun ULongProgression.first(): ULong {\n    if (isEmpty())\n        throw

```

```

NoSuchElementException("Progression $this is empty.")\n    return this.first\n}\n\n/**\n * Returns the first
element, or `null` if the progression is empty.\n */\n@SinceKotlin("1.7")\npublic fun
UIntProgression.firstOrNull(): UInt? {\n    return if (isEmpty()) null else this.first\n}\n\n/**\n * Returns the first
element, or `null` if the progression is empty.\n */\n@SinceKotlin("1.7")\npublic fun
ULongProgression.firstOrNull(): ULong? {\n    return if (isEmpty()) null else this.first\n}\n\n/**\n * Returns the last
element.\n */\n * @throws NoSuchElementException if the progression is empty.\n */\n * @sample
samples.collections.Collections.Elements.last\n */\n@SinceKotlin("1.7")\npublic fun UIntProgression.last(): UInt
{\n    if (isEmpty())\n        throw NoSuchElementException("Progression $this is empty.")\n    return
this.last\n}\n\n/**\n * Returns the last element.\n */\n * @throws NoSuchElementException if the progression is
empty.\n */\n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin("1.7")\npublic fun
ULongProgression.last(): ULong {\n    if (isEmpty())\n        throw NoSuchElementException("Progression $this is
empty.")\n    return this.last\n}\n\n/**\n * Returns the last element, or `null` if the progression is empty.\n */\n *
@sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin("1.7")\npublic fun
UIntProgression.lastOrNull(): UInt? {\n    return if (isEmpty()) null else this.last\n}\n\n/**\n * Returns the last
element, or `null` if the progression is empty.\n */\n * @sample samples.collections.Collections.Elements.last\n
*/\n@SinceKotlin("1.7")\npublic fun ULongProgression.lastOrNull(): ULong? {\n    return if (isEmpty()) null else
this.last\n}\n\n/**\n * Returns a random element from this range.\n */\n * @throws IllegalArgumentException if this
range is empty.\n */\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\
npublic inline fun UIntRange.random(): UInt {\n    return random(Random)\n}\n\n/**\n * Returns a random element
from this range.\n */\n * @throws IllegalArgumentException if this range is empty.\n */\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\
npublic inline fun ULongRange.random(): ULong {\n    return random(Random)\n}\n\n/**\n * Returns a random
element from this range using the specified source of randomness.\n */\n * @throws IllegalArgumentException if
this range is empty.\n */\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic
fun UIntRange.random(random: Random): UInt {\n    try {\n        return random.nextUInt(this)\n    } catch(e:
IllegalArgumentException) {\n        throw NoSuchElementException(e.message)\n    }\n}\n\n/**\n * Returns a
random element from this range using the specified source of randomness.\n */\n * @throws
IllegalArgumentException if this range is empty.\n */\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
ULongRange.random(random: Random): ULong {\n    try {\n        return random.nextULong(this)\n    } catch(e:
IllegalArgumentException) {\n        throw NoSuchElementException(e.message)\n    }\n}\n\n/**\n * Returns a
random element from this range, or `null` if this range is empty.\n */\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UIntRange.randomOrNull():
UInt? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this range, or `null` if this
range is empty.\n */\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULongRange.randomOrNull():
ULong? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this range using the
specified source of randomness, or `null` if this range is empty.\n */\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\npublic fun UIntRange.randomOrNull(random: Random): UInt? {\n    if
(isEmpty())\n        return null\n    return random.nextUInt(this)\n}\n\n/**\n * Returns a random element from this
range using the specified source of randomness, or `null` if this range is empty.\n */\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\npublic fun ULongRange.randomOrNull(random: Random): ULong? {\n    if
(isEmpty())\n        return null\n    return random.nextULong(this)\n}\n\n/**\n * Returns `true` if this range contains
the specified [element].\n */\n * Always returns `false` if the [element] is `null`.\n */

```

```

*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline operator fun UIntRange.contains(element: UInt?): Boolean {\n    return element != null &&\n    contains(element)\n}\n\n/**\n * Returns `true` if this range contains the specified [element].\n * \n * Always returns `false` if the [element] is `null`.\n */\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline operator fun ULongRange.contains(element: ULong?): Boolean {\n    return element != null &&\n    contains(element)\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic operator fun\nUIntRange.contains(value: UByte): Boolean {\n    return contains(value.toInt())\n}\n\n/**\n * Checks if the\nspecified [value] belongs to this range.\n */\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic operator fun\nULongRange.contains(value: UByte): Boolean {\n    return contains(value.toULong())\n}\n\n/**\n * Checks if the\nspecified [value] belongs to this range.\n */\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic operator fun\nULongRange.contains(value: UInt): Boolean {\n    return contains(value.toULong())\n}\n\n/**\n * Checks if the\nspecified [value] belongs to this range.\n */\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic operator fun\nUIntRange.contains(value: ULong): Boolean {\n    return (value shr UInt.SIZE_BITS) == 0uL &&\n    contains(value.toInt())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic operator fun\nUIntRange.contains(value: UShort): Boolean {\n    return contains(value.toInt())\n}\n\n/**\n * Checks if the\nspecified [value] belongs to this range.\n */\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic operator fun\nULongRange.contains(value: UShort): Boolean {\n    return contains(value.toULong())\n}\n\n/**\n * Returns a\nprogression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * \n * If the [to] value is greater than `this` value the returned progression is empty.\n */\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun\nUByte.downTo(to: UByte): UIntProgression {\n    return UIntProgression.fromClosedRange(this.toInt(),\n    to.toInt(), -1)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -\n1.\n * \n * The [to] value should be less than or equal to `this` value.\n * \n * If the [to] value is greater than `this` value\nthe returned progression is empty.\n */\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun\nUInt.downTo(to: UInt): UIntProgression {\n    return UIntProgression.fromClosedRange(this, to, -1)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * \n * If the [to] value is greater than `this` value the returned progression is\nempty.\n */\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun\nULong.downTo(to: ULong): ULongProgression {\n    return ULongProgression.fromClosedRange(this, to, -\n1L)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * \n * If the [to] value is greater than `this` value the\nreturned progression is empty.\n */\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun\nUShort.downTo(to: UShort): UIntProgression {\n    return UIntProgression.fromClosedRange(this.toInt(),\n    to.toInt(), -1)\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If\nthe [to] value is less than or equal to `this` value, then the returned range is empty.\n */\n\n*\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline operator fun\nUByte.rangeUntil(to: UByte): UIntRange {\n    return until(to)\n}\n\n/**\n * Returns a range from this value up to\nbut excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned\nrange is empty.\n */\n\n*\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline

```

```

operator fun UInt.rangeUntil(to: UInt): UIntRange { \n  return until(to)\n}\n\n/**\n * Returns a range from this
value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then
the returned range is empty.\n
*\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline operator fun
ULong.rangeUntil(to: ULong): ULongRange { \n  return until(to)\n}\n\n/**\n * Returns a range from this value up
to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the
returned range is empty.\n
*\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline operator fun
UShort.rangeUntil(to: UShort): UIntRange { \n  return until(to)\n}\n\n/**\n * Returns a progression that goes over
the same range in the opposite direction with the same step.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UIntProgression.reversed(): UIntProgression { \n  return UIntProgression.fromClosedRange(last, first, -
step)\n}\n\n/**\n * Returns a progression that goes over the same range in the opposite direction with the same
step.\n *\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
ULongProgression.reversed(): ULongProgression { \n  return ULongProgression.fromClosedRange(last, first, -
step)\n}\n\n/**\n * Returns a progression that goes over the same range with the given step.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun
UIntProgression.step(step: Int): UIntProgression { \n  checkStepIsPositive(step > 0, step)\n  return
UIntProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)\n}\n\n/**\n * Returns a progression
that goes over the same range with the given step.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun
ULongProgression.step(step: Long): ULongProgression { \n  checkStepIsPositive(step > 0, step)\n  return
ULongProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)\n}\n\n/**\n * Returns a range from
this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value,
then the returned range is empty.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun
UByte.until(to: UByte): UIntRange { \n  if (to <= UByte.MIN_VALUE) return UIntRange.EMPTY\n  return
this.toUInt() .. (to - 1u).toUInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to]
value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun UInt.until(to:
UInt): UIntRange { \n  if (to <= UInt.MIN_VALUE) return UIntRange.EMPTY\n  return this .. (to -
1u).toUInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the
[to] value is less than or equal to `this` value, then the returned range is empty.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun
ULong.until(to: ULong): ULongRange { \n  if (to <= ULong.MIN_VALUE) return ULongRange.EMPTY\n
return this .. (to - 1u).toULong()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to]
value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun
UShort.until(to: UShort): UIntRange { \n  if (to <= UShort.MIN_VALUE) return UIntRange.EMPTY\n  return
this.toUInt() .. (to - 1u).toUInt()\n}\n\n/**\n * Ensures that this value is not less than the specified
[minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the
[minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeastUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UInt.coerceAtLeast(minimumValue: UInt): UInt { \n  return if (this < minimumValue) minimumValue else
this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value
if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtLeastUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun

```

```

ULong.coerceAtLeast(minimumValue: ULong): ULong {\n  return if (this < minimumValue) minimumValue else
this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value
if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtLeastUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UByte.coerceAtLeast(minimumValue: UByte): UByte {\n  return if (this < minimumValue) minimumValue else
this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value
if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtLeastUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UShort.coerceAtLeast(minimumValue: UShort): UShort {\n  return if (this < minimumValue) minimumValue else
this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this
value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMostUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UInt.coerceAtMost(maximumValue: UInt): UInt {\n  return if (this > maximumValue) maximumValue else
this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this
value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMostUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
ULong.coerceAtMost(maximumValue: ULong): ULong {\n  return if (this > maximumValue) maximumValue else
this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this
value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMostUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UByte.coerceAtMost(maximumValue: UByte): UByte {\n  return if (this > maximumValue) maximumValue else
this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this
value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMostUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UShort.coerceAtMost(maximumValue: UShort): UShort {\n  return if (this > maximumValue) maximumValue
else this\n}\n\n/**\n * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].\n * \n
* @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or
[maximumValue] if this value is greater than [maximumValue].\n * \n * @sample
samples.comparisons.ComparableOps.coerceInUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UInt.coerceIn(minimumValue: UInt, maximumValue: UInt): UInt {\n  if (minimumValue > maximumValue)
throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than
minimum $minimumValue.")\n  if (this < minimumValue) return minimumValue\n  if (this > maximumValue)
return maximumValue\n  return this\n}\n\n/**\n * Ensures that this value lies in the specified range
[minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value
is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n * @sample
samples.comparisons.ComparableOps.coerceInUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
ULong.coerceIn(minimumValue: ULong, maximumValue: ULong): ULong {\n  if (minimumValue >
maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum
$maximumValue is less than minimum $minimumValue.")\n  if (this < minimumValue) return minimumValue\n
if (this > maximumValue) return maximumValue\n  return this\n}\n\n/**\n * Ensures that this value lies in the

```

```

specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or
[minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than
[maximumValue].\n * \n * @sample samples.comparisons.ComparableOps.coerceInUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UByte.coerceIn(minimumValue: UByte, maximumValue: UByte): UByte {\n if (minimumValue >
maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum
$maximumValue is less than minimum $minimumValue.")\n if (this < minimumValue) return minimumValue\n
if (this > maximumValue) return maximumValue\n return this\n}\n\n/**\n * Ensures that this value lies in the
specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or
[minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than
[maximumValue].\n * \n * @sample samples.comparisons.ComparableOps.coerceInUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UShort.coerceIn(minimumValue: UShort, maximumValue: UShort): UShort {\n if (minimumValue >
maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum
$maximumValue is less than minimum $minimumValue.")\n if (this < minimumValue) return minimumValue\n
if (this > maximumValue) return maximumValue\n return this\n}\n\n/**\n * Ensures that this value lies in the
specified [range].\n * \n * @return this value if it's in the [range], or `range.start` if this value is less than
`range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`.\n * \n * @sample
samples.comparisons.ComparableOps.coerceInUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UInt.coerceIn(range: ClosedRange<UInt>): UInt {\n if (range is ClosedFloatingPointRange) {\n return
this.coerceIn<UInt>(range)\n }\n if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to
an empty range: $range.")\n return when {\n this < range.start -> range.start\n this > range.endInclusive -
-> range.endInclusive\n else -> this\n }\n}\n\n/**\n * Ensures that this value lies in the specified [range].\n * \n *
@return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or
`range.endInclusive` if this value is greater than `range.endInclusive`.\n * \n * @sample
samples.comparisons.ComparableOps.coerceInUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
ULong.coerceIn(range: ClosedRange<ULong>): ULong {\n if (range is ClosedFloatingPointRange) {\n return
this.coerceIn<ULong>(range)\n }\n if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value
to an empty range: $range.")\n return when {\n this < range.start -> range.start\n this >
range.endInclusive -> range.endInclusive\n else -> this\n }\n}\n\n"/*\n * Copyright 2010-2022 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n
*\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("USequencesKt")\n\npackage
kotlin.sequences\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.*\n\n/**\n * Returns the
sum of all elements in the sequence.\n * \n * The operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfUInt")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedT
ypes::class)\npublic fun Sequence<UInt>.sum(): UInt {\n var sum: UInt = 0u\n for (element in this) {\n sum
+= element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n * \n * The
operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfULong")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Sequence<ULong>.sum(): ULong {\n var sum: ULong = 0uL\n for (element in this)
{\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n * \n *
The operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfUByte")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Sequence<UByte>.sum(): UInt {\n var sum: UInt = 0u\n for (element in this) {\n

```

```

sum += element\n } \n return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n * \n * The
operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("\sumOfUShort")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Sequence<UShort>.sum(): UInt { \n var sum: UInt = 0u\n for (element in this) { \n
sum += element\n } \n return sum\n}\n\n"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n * \n\npackage kotlin\n\npublic expect open class Error : Throwable { \n
constructor()\n constructor(message: String?)\n constructor(message: String?, cause: Throwable?)\n
constructor(cause: Throwable?)\n}\n\npublic expect open class Exception : Throwable { \n constructor()\n
constructor(message: String?)\n constructor(message: String?, cause: Throwable?)\n constructor(cause:
Throwable?)\n}\n\npublic expect open class RuntimeException : Exception { \n constructor()\n
constructor(message: String?)\n constructor(message: String?, cause: Throwable?)\n constructor(cause:
Throwable?)\n}\n\npublic expect open class IllegalArgumentException : RuntimeException { \n constructor()\n
constructor(message: String?)\n constructor(message: String?, cause: Throwable?)\n constructor(cause:
Throwable?)\n}\n\npublic expect open class IllegalStateException : RuntimeException { \n constructor()\n
constructor(message: String?)\n constructor(message: String?, cause: Throwable?)\n constructor(cause:
Throwable?)\n}\n\npublic expect open class IndexOutOfBoundsException : RuntimeException { \n constructor()\n
constructor(message: String?)\n}\n\npublic expect open class ConcurrentModificationException :
RuntimeException { \n constructor()\n constructor(message: String?)\n @Deprecated("\The constructor is not
supported on all platforms and will be removed from kotlin-stdlib-common soon.", level =
DeprecationLevel.ERROR)\n constructor(message: String?, cause: Throwable?)\n @Deprecated("\The
constructor is not supported on all platforms and will be removed from kotlin-stdlib-common soon.", level =
DeprecationLevel.ERROR)\n constructor(cause: Throwable?)\n}\n\npublic expect open class
UnsupportedOperationException : RuntimeException { \n constructor()\n constructor(message: String?)\n
constructor(message: String?, cause: Throwable?)\n constructor(cause: Throwable?)\n}\n\npublic expect open
class NumberFormatException : IllegalArgumentException { \n constructor()\n constructor(message:
String?)\n}\n\npublic expect open class NullPointerException : RuntimeException { \n constructor()\n
constructor(message: String?)\n}\n\npublic expect open class ClassCastException : RuntimeException { \n
constructor()\n constructor(message: String?)\n}\n\npublic expect open class AssertionError : Error { \n
constructor()\n constructor(message: Any?)\n}\n\npublic expect open class NoSuchElementException :
RuntimeException { \n constructor()\n constructor(message: String?)\n}\n\n@SinceKotlin("1.3")\npublic
expect open class ArithmeticException : RuntimeException { \n constructor()\n constructor(message:
String?)\n}\n\n@Deprecated("\This exception type is not supposed to be thrown or caught in common code and will
be removed from kotlin-stdlib-common soon.", level = DeprecationLevel.ERROR)\npublic expect open class
NoWhenBranchMatchedException : RuntimeException { \n constructor()\n constructor(message: String?)\n
constructor(message: String?, cause: Throwable?)\n constructor(cause: Throwable?)\n}\n\n@Deprecated("\This
exception type is not supposed to be thrown or caught in common code and will be removed from kotlin-stdlib-
common soon.", level = DeprecationLevel.ERROR)\npublic expect class UninitializedPropertyAccessException :
RuntimeException { \n constructor()\n constructor(message: String?)\n constructor(message: String?, cause:
Throwable?)\n constructor(cause: Throwable?)\n}\n\n/**\n * Thrown after invocation of a function or property
that was expected to return `Nothing`, but returned something instead.\n
*\n@SinceKotlin("1.4")\n@PublishedApi\ninternal class KotlinNothingValueException : RuntimeException { \n
constructor() : super()\n constructor(message: String?) : super(message)\n constructor(message: String?, cause:
Throwable?) : super(message, cause)\n constructor(cause: Throwable?) : super(cause)\n}\n\n\n/**\n * Returns the
detailed description of this throwable with its stack trace.\n * \n * The detailed description includes:\n * - the short
description (see [Throwable.toString]) of this throwable;\n * - the complete stack trace;\n * - detailed descriptions of
the exceptions that were [suppressed][suppressedExceptions] in order to deliver this exception;\n * - the detailed
description of each throwable in the [Throwable.cause] chain.\n * \n\n@SinceKotlin("1.4")\npublic expect fun

```


Throwable.stackTraceToString(): String\n\n/**\n * Prints the [detailed description][Throwable.stackTraceToString] of this throwable to the standard output or standard error output.\n

*\n@SinceKotlin("1.4")\n@Suppress("EXTENSION_SHADOWED_BY_MEMBER")\npublic expect fun Throwable.printStackTrace(): Unit\n\n/**\n * When supported by the platform, adds the specified exception to the list of exceptions that were\n * suppressed in order to deliver this exception.\n

*\n@SinceKotlin("1.4")\n@Suppress("EXTENSION_SHADOWED_BY_MEMBER")\npublic expect fun Throwable.addSuppressed(exception: Throwable)\n\n/**\n * Returns a list of all exceptions that were suppressed in order to deliver this exception.\n * The list can be empty:\n * - if no exceptions were suppressed;\n * - if the platform doesn't support suppressed exceptions;\n * - if this [Throwable] instance has disabled the suppression.\n

\n@SinceKotlin("1.4")\npublic expect val Throwable.suppressedExceptions: List<Throwable>\n","/\n

Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\npackage kotlin.js\nimport kotlin.annotation.AnnotationTarget.*\n\n/**\n * Gives a declaration (a function, a property or a class) specific name in JavaScript.\n */\n@Target(CLASS, FUNCTION, PROPERTY, CONSTRUCTOR, PROPERTY_GETTER, PROPERTY_SETTER)\n@OptionalExpectation\npublic expect annotation class JsName(val name: String)\n\n/**\n * Marks experimental JS export annotations.\n * Note that behavior of these annotations will likely be changed in the future.\n * Usages of such annotations will be reported as warnings unless an explicit opt-in with\n * the [OptIn] annotation, e.g. `@OptIn(ExperimentalJsExport::class)`,\n * or with the `--opt-in=kotlin.js.ExperimentalJsExport` compiler option is given.\n */\n@RequiresOptIn(level = RequiresOptIn.Level.WARNING)\n@MustBeDocumented\n@Retention(AnnotationRetention.BINARY)\n@SinceKotlin("1.4")\npublic annotation class ExperimentalJsExport\n\n/**\n * Exports top-level declaration on JS platform.\n * Compiled module exposes declarations that are marked with this annotation without name mangling.\n * This annotation can be applied to either files or top-level declarations.\n * It is currently prohibited to export the following kinds of declarations:\n * * `expect` declarations\n * * inline functions with reified type parameters\n * * suspend functions\n * * secondary constructors without `@JsName`\n * * extension properties\n * * enum classes\n * * annotation classes\n * Signatures of exported declarations must only contain "exportable" types:\n * * `dynamic`, `Any`, `String`, `Boolean`, `Byte`, `Short`, `Int`, `Float`, `Double`\n * * `BooleanArray`, `ByteArray`, `ShortArray`, `IntArray`, `FloatArray`, `DoubleArray`\n * * `Array<exportable-type>`\n * * Function types with exportable parameters and return types\n * * `external` or `@JsExport` classes and interfaces\n * * Nullable counterparts of types above\n * * Unit return type. Must not be nullable\n * This annotation is experimental, meaning that restrictions mentioned above are subject to change.\n */\n@ExperimentalJsExport\n@Retention(AnnotationRetention.BINARY)\n@Target(CLASS, PROPERTY, FUNCTION, FILE)\n@SinceKotlin("1.4")\n@OptionalExpectation\npublic expect annotation class JsExport(),"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\npackage kotlin.io\n\n/**\n * Prints the line separator to the standard output stream.\n */\npublic expect fun println()\n\n/**\n * Prints the given [message] and the line separator to the standard output stream.\n */\npublic expect fun println(message: Any?)\n\n/**\n * Prints the given [message] to the standard output stream.\n */\npublic expect fun print(message: Any?)\n\n/**\n * Reads a line of input from the standard input stream and returns it,\n * or throws a [RuntimeException] if EOF has already been reached when [readLn] is called.\n * LF or CRLF is treated as the line terminator. Line terminator is not included in the returned string.\n * Currently this function is not supported in Kotlin/JS and throws [UnsupportedOperationException].\n */\n@SinceKotlin("1.6")\npublic expect fun readLn(): String\n\n/**\n * Reads a line of input from the standard input stream and returns it,\n * or return `null` if EOF has already been reached when [readLnOrNull] is called.\n * LF or CRLF is treated as the line terminator. Line terminator is not included in the returned string.\n * Currently this function is not supported in Kotlin/JS and throws [UnsupportedOperationException].\n */\n@SinceKotlin("1.6")\npublic expect fun readLnOrNull(): String?\n\ninternal class ReadAfterEOFException(message: String?) : RuntimeException(message)\n\ninternal expect interface Serializable\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language

contributors.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

```

package kotlin.collections
import kotlin.internal.PlatformDependent

Classes that inherit from this interface can be represented as a sequence of elements that can
be iterated over.
@param T the type of element being iterated over. The iterator is covariant in its element type.
public interface Iterable<out T> {
    /**
     * Returns an iterator over the elements of this object.
     */
    public operator fun iterator(): Iterator<T>
}

Classes that inherit from this interface can be represented as a sequence of
elements that can be iterated over and that supports removing elements during iteration.
@param T the type
of element being iterated over. The mutable iterator is invariant in its element type.
public interface MutableIterable<out T> : Iterable<T> {
    /**
     * Returns an iterator over the elements of this sequence that
     supports removing elements during iteration.
     */
    override fun iterator(): MutableIterator<T>
}

A generic collection of elements. Methods in this interface support only read-only access to the collection;
read/write access is supported through the [MutableCollection] interface.
@param E the type of elements
contained in the collection. The collection is covariant in its element type.
public interface Collection<out E> :
Iterable<E> {
    // Query Operations
    /**
     * Returns the size of the collection.
     */
    public val size: Int
    /**
     * Returns `true` if the collection is empty (contains no elements), `false` otherwise.
     */
    public fun isEmpty(): Boolean
    /**
     * Checks if the specified element is contained in this collection.
     */
    public operator fun contains(element: @UnsafeVariance E): Boolean
    override fun iterator():
Iterator<E>
    // Bulk Operations
    /**
     * Checks if all elements in the specified collection are contained in
     this collection.
     */
    public fun containsAll(elements: Collection<@UnsafeVariance E>): Boolean
}

A generic collection of elements that supports adding and removing elements.
@param E the type of
elements contained in the collection. The mutable collection is invariant in its element type.
public interface MutableCollection<E> : Collection<E>, MutableIterable<E> {
    // Query Operations
    override fun iterator():
MutableIterator<E>
    // Modification Operations
    /**
     * Adds the specified element to the collection.
     */
    /**
     * @return `true` if the element has been added, `false` if the collection does not support duplicates
     and
     the element is already contained in the collection.
     */
    public fun add(element: E): Boolean
    /**
     * Removes a single instance of the specified element from this
     collection, if it is present.
     */
    /**
     * @return
     `true` if the element has been successfully removed; `false` if it was not present in the collection.
     */
    public fun remove(element: E): Boolean
    // Bulk Modification Operations
    /**
     * Adds all of the elements of
     the specified collection to this collection.
     */
    /**
     * @return `true` if any of the specified elements was added to
     the collection, `false` if the collection was not modified.
     */
    public fun addAll(elements: Collection<E>):
Boolean
    /**
     * Removes all of this collection's elements that are also contained in the specified
     collection.
     */
    /**
     * @return `true` if any of the specified elements was removed from the collection, `false` if
     the collection was not modified.
     */
    public fun removeAll(elements: Collection<E>): Boolean
    /**
     * Retains only the elements in this collection that are contained in the specified collection.
     */
    /**
     * @return
     `true` if any element was removed from the collection, `false` if the collection was not modified.
     */
    public fun retainAll(elements: Collection<E>): Boolean
    /**
     * Removes all elements from this collection.
     */
    public fun clear(): Unit
}

A generic ordered collection of elements. Methods in this interface
support only read-only access to the list;
read/write access is supported through the [MutableList] interface.
@param E the type of elements contained in the list. The list is covariant in its element type.
public interface List<out E> : Collection<E> {
    // Query Operations
    override val size: Int
    override fun isEmpty():
Boolean
    override fun contains(element: @UnsafeVariance E): Boolean
    override fun iterator():
Iterator<E>
    // Bulk Operations
    override fun containsAll(elements: Collection<@UnsafeVariance E>):
Boolean
    // Positional Access Operations
    /**
     * Returns the element at the specified index in the list.
     */
    public operator fun get(index: Int): E
    // Search Operations
    /**
     * Returns the index of the first
     occurrence of the specified element in the list, or -1 if the specified
     element is not contained in the list.
     */
    /**
     * @return
     the index of the first occurrence of the specified element in the list, or -1 if the specified
     element is not contained in the list.
     */
    public fun indexOf(element: @UnsafeVariance E): Int
    /**
     * Returns the index of the last
     occurrence of the specified element in the list, or -1 if the specified
     element is not contained in the list.
     */
    /**
     * @return
     the index of the last occurrence of the specified element in the list, or -1 if the specified
     element is not contained in the list.
     */
    public fun lastIndexOf(element: @UnsafeVariance E): Int
    // List Iterators
    /**
     * Returns a list

```

```

iterator over the elements in this list (in proper sequence).\n    */\n    public fun listIterator(): ListIterator<E>\n\n/**\n * Returns a list iterator over the elements in this list (in proper sequence), starting at the specified [index].\n *\n */\n    public fun listIterator(index: Int): ListIterator<E>\n    // View\n    /**\n * Returns a view of the portion of this list between the specified [fromIndex] (inclusive) and [toIndex] (exclusive).\n * The returned list is backed by this list, so non-structural changes in the returned list are reflected in this list, and vice-versa.\n *\n * Structural changes in the base list make the behavior of the view undefined.\n */\n    public fun subList(fromIndex: Int, toIndex: Int): List<E>\n\n/**\n * A generic ordered collection of elements that supports adding and removing elements.\n * @param E the type of elements contained in the list. The mutable list is invariant in its element type.\n */\npublic interface MutableList<E> : List<E>, MutableCollection<E> {\n    // Modification Operations\n    /**\n * Adds the specified element to the end of this list.\n *\n * @return `true` because the list is always modified as the result of this operation.\n */\n    override fun add(element: E): Boolean\n\n    override fun remove(element: E): Boolean\n\n    // Bulk Modification Operations\n    /**\n * Adds all of the elements of the specified collection to the end of this list.\n *\n * The elements are appended in the order they appear in the [elements] collection.\n *\n * @return `true` if the list was changed as the result of the operation.\n */\n    override fun addAll(elements: Collection<E>): Boolean\n\n    /**\n * Inserts all of the elements of the specified collection [elements] into this list at the specified [index].\n *\n * @return `true` if the list was changed as the result of the operation.\n */\n    public fun addAll(index: Int, elements: Collection<E>): Boolean\n\n    override fun removeAll(elements: Collection<E>): Boolean\n\n    override fun retainAll(elements: Collection<E>): Boolean\n\n    override fun clear(): Unit\n\n    // Positional Access Operations\n    /**\n * Replaces the element at the specified position in this list with the specified element.\n *\n * @return the element previously at the specified position.\n */\n    public operator fun set(index: Int, element: E): E\n\n    /**\n * Inserts an element into the list at the specified [index].\n */\n    public fun add(index: Int, element: E): Unit\n\n    /**\n * Removes an element at the specified [index] from the list.\n *\n * @return the element that has been removed.\n */\n    public fun removeAt(index: Int): E\n\n    // List Iterators\n    override fun listIterator(): MutableListIterator<E>\n\n    override fun listIterator(index: Int): MutableListIterator<E>\n\n    // View\n    override fun subList(fromIndex: Int, toIndex: Int): MutableList<E>\n\n/**\n * A generic unordered collection of elements that does not support duplicate elements.\n * Methods in this interface support only read-only access to the set;\n * read/write access is supported through the [MutableSet] interface.\n * @param E the type of elements contained in the set. The set is covariant in its element type.\n */\npublic interface Set<out E> : Collection<E> {\n    // Query Operations\n\n    override val size: Int\n\n    override fun isEmpty(): Boolean\n\n    override fun contains(element: @UnsafeVariance E): Boolean\n\n    override fun iterator(): Iterator<E>\n\n    // Bulk Operations\n    override fun containsAll(elements: Collection<@UnsafeVariance E>): Boolean\n\n/**\n * A generic unordered collection of elements that does not support duplicate elements, and supports\n * adding and removing elements.\n * @param E the type of elements contained in the set. The mutable set is invariant in its element type.\n */\npublic interface MutableSet<E> : Set<E>, MutableCollection<E> {\n    // Query Operations\n    override fun iterator(): MutableIterator<E>\n\n    // Modification Operations\n\n    /**\n * Adds the specified element to the set.\n *\n * @return `true` if the element has been added, `false` if the element is already contained in the set.\n */\n    override fun add(element: E): Boolean\n\n    override fun remove(element: E): Boolean\n\n    // Bulk Modification Operations\n\n    override fun addAll(elements: Collection<E>): Boolean\n\n    override fun removeAll(elements: Collection<E>): Boolean\n\n    override fun retainAll(elements: Collection<E>): Boolean\n\n    override fun clear(): Unit\n\n/**\n * A collection that holds pairs of objects (keys and values) and supports efficiently retrieving\n * the value corresponding to each key. Map keys are unique; the map holds only one value for each key.\n * Methods in this interface support only read-only access to the map; read-write access is supported through\n * the [MutableMap] interface.\n * @param K the type of map keys. The map is invariant in its key type, as it\n * can accept key as a parameter (of [containsKey] for example) and return it in [keys] set.\n * @param V the type of map values. The map is covariant in its value type.\n */\npublic interface Map<K, out V> {\n    // Query Operations\n\n    /**\n * Returns the number of key/value pairs in the map.\n */\n    public val size: Int\n\n    /**\n * Returns `true` if the map is empty (contains no elements), `false` otherwise.\n */\n    public fun isEmpty(): Boolean\n
```

```

/**\n * Returns `true` if the map contains the specified [key].\n */\n public fun containsKey(key: K):
Boolean\n\n /**\n * Returns `true` if the map maps one or more keys to the specified [value].\n */\n public
fun containsValue(value: @UnsafeVariance V): Boolean\n\n /**\n * Returns the value corresponding to the
given [key], or `null` if such a key is not present in the map.\n */\n public operator fun get(key: K): V?\n\n
/**\n * Returns the value corresponding to the given [key], or [defaultValue] if such a key is not present in the
map.\n */\n * @since JDK 1.8\n */\n @SinceKotlin("1.1")\n @PlatformDependent\n public fun
getOrDefault(key: K, defaultValue: @UnsafeVariance V): V {\n // See default implementation in JDK
sources\n throw NotImplementedError()\n }\n\n // Views\n /**\n * Returns a read-only [Set] of all keys
in this map.\n */\n public val keys: Set<K>\n\n /**\n * Returns a read-only [Collection] of all values in this
map. Note that this collection may contain duplicate values.\n */\n public val values: Collection<V>\n\n /**\n
* Returns a read-only [Set] of all key/value pairs in this map.\n */\n public val entries: Set<Map.Entry<K,
V>>\n\n /**\n * Represents a key/value pair held by a [Map].\n */\n public interface Entry<out K, out V>
{\n /**\n * Returns the key of this key/value pair.\n */\n public val key: K\n\n /**\n *
Returns the value of this key/value pair.\n */\n public val value: V\n }\n\n /**\n * A modifiable
collection that holds pairs of objects (keys and values) and supports efficiently retrieving\n * the value
corresponding to each key. Map keys are unique; the map holds only one value for each key.\n * @param K the type
of map keys. The map is invariant in its key type.\n * @param V the type of map values. The mutable map is
invariant in its value type.\n */\n public interface MutableMap<K, V> : Map<K, V> {\n // Modification
Operations\n /**\n * Associates the specified [value] with the specified [key] in the map.\n */\n * @return
the previous value associated with the key, or `null` if the key was not present in the map.\n */\n public fun
put(key: K, value: V): V?\n\n /**\n * Removes the specified key and its corresponding value from this map.\n
*/\n * @return the previous value associated with the key, or `null` if the key was not present in the map.\n
*/\n public fun remove(key: K): V?\n\n /**\n * Removes the entry for the specified key only if it is mapped to the
specified value.\n */\n * @return true if entry was removed\n */\n @SinceKotlin("1.1")\n
@PlatformDependent\n public fun remove(key: K, value: V): Boolean {\n // See default implementation in
JDK sources\n return true\n }\n\n // Bulk Modification Operations\n /**\n * Updates this map with
key/value pairs from the specified map [from].\n */\n public fun putAll(from: Map<out K, V>): Unit\n\n /**\n
* Removes all elements from this map.\n */\n public fun clear(): Unit\n\n // Views\n /**\n * Returns a
[MutableSet] of all keys in this map.\n */\n override val keys: MutableSet<K>\n\n /**\n * Returns a
[MutableCollection] of all values in this map. Note that this collection may contain duplicate values.\n */\n
override val values: MutableCollection<V>\n\n /**\n * Returns a [MutableSet] of all key/value pairs in this
map.\n */\n override val entries: MutableSet<MutableMap.MutableEntry<K, V>>\n\n /**\n * Represents a
key/value pair held by a [MutableMap].\n */\n public interface MutableEntry<K, V> : Map.Entry<K, V> {\n
/**\n * Changes the value associated with the key of this entry.\n */\n * @return the previous value
corresponding to the key.\n */\n public fun setValue(newValue: V): V\n }\n\n "/*\n * Copyright 2010-
2015 JetBrains s.r.o.\n * Licensed under the Apache License, Version 2.0 (the "License");\n * you may not use
this file except in compliance with the License.\n * You may obtain a copy of the License at\n *
http://www.apache.org/licenses/LICENSE-2.0\n * Unless required by applicable law or agreed to in writing,
software\n * distributed under the License is distributed on an "AS IS" BASIS,\n * WITHOUT WARRANTIES
OR CONDITIONS OF ANY KIND, either express or implied.\n * See the License for the specific language
governing permissions and\n * limitations under the License.\n */\n\n package kotlin\n\n /**\n * The type with only
one value: the `Unit` object. This type corresponds to the `void` type in Java.\n */\n public object Unit {\n
override fun toString() = "kotlin.Unit"\n }\n\n "/*\n * Copyright 2010-2015 JetBrains s.r.o.\n * Licensed under the
Apache License, Version 2.0 (the "License");\n * you may not use this file except in compliance with the
License.\n * You may obtain a copy of the License at\n *
http://www.apache.org/licenses/LICENSE-2.0\n * Unless required by applicable law or agreed to in writing,
software\n * distributed under the License is distributed on an "AS IS" BASIS,\n * WITHOUT WARRANTIES
OR CONDITIONS OF ANY KIND, either express or implied.\n * See the License for the specific language governing
permissions and\n * limitations under the

```

```

License.\n */\n\npackage kotlin.annotation\n\nimport kotlin.annotation.AnnotationTarget.*\n\n/**\n * Contains the list of code elements which are the possible annotation targets\n */\npublic enum class AnnotationTarget {\n    /**\n     * Class, interface or object, annotation class is also included\n     */\n    CLASS,\n    /**\n     * Annotation class only\n     */\n    ANNOTATION_CLASS,\n    /**\n     * Generic type parameter\n     */\n    TYPE_PARAMETER,\n    /**\n     * Property\n     */\n    PROPERTY,\n    /**\n     * Field, including property's backing field\n     */\n    FIELD,\n    /**\n     * Local variable\n     */\n    LOCAL_VARIABLE,\n    /**\n     * Value parameter of a function or a constructor\n     */\n    VALUE_PARAMETER,\n    /**\n     * Constructor only (primary or secondary)\n     */\n    CONSTRUCTOR,\n    /**\n     * Function (constructors are not included)\n     */\n    FUNCTION,\n    /**\n     * Property getter only\n     */\n    PROPERTY_GETTER,\n    /**\n     * Property setter only\n     */\n    PROPERTY_SETTER,\n    /**\n     * Type usage\n     */\n    TYPE,\n    /**\n     * Any expression\n     */\n    EXPRESSION,\n    /**\n     * File\n     */\n    FILE,\n    /**\n     * Type alias\n     */\n    @SinceKotlin("1.1")\n    TYPEALIAS\n}\n\n/**\n * Contains the list of possible annotation's retentions.\n */\n * Determines how an annotation is stored in binary output.\n */\npublic enum class AnnotationRetention {\n    /**\n     * Annotation isn't stored in binary output\n     */\n    SOURCE,\n    /**\n     * Annotation is stored in binary output, but invisible for reflection\n     */\n    BINARY,\n    /**\n     * Annotation is stored in binary output and visible for reflection (default retention)\n     */\n    RUNTIME\n}\n\n/**\n * This meta-annotation indicates the kinds of code elements which are possible targets of an annotation.\n */\n * If the target meta-annotation is not present on an annotation declaration, the annotation is applicable to the following elements:\n * [CLASS], [PROPERTY], [FIELD], [LOCAL_VARIABLE], [VALUE_PARAMETER], [CONSTRUCTOR], [FUNCTION], [PROPERTY_GETTER], [PROPERTY_SETTER].\n */\n * @property allowedTargets list of allowed annotation targets\n */\n * @Target(AnnotationTarget.ANNOTATION_CLASS)\n */\npublic annotation class Target(vararg val allowedTargets: AnnotationTarget)\n\n/**\n * This meta-annotation determines whether an annotation is stored in binary output and visible for reflection. By default, both are true.\n */\n * @property value necessary annotation retention (RUNTIME, BINARY or SOURCE)\n */\n * @Target(AnnotationTarget.ANNOTATION_CLASS)\n */\npublic annotation class Retention(val value: AnnotationRetention = AnnotationRetention.RUNTIME)\n\n/**\n * This meta-annotation determines that an annotation is applicable twice or more on a single code element\n */\n * @Target(AnnotationTarget.ANNOTATION_CLASS)\n */\npublic annotation class Repeatable\n\n/**\n * This meta-annotation determines that an annotation is a part of public API and therefore should be included in the generated\n * documentation for the element to which the annotation is applied.\n */\n * @Target(AnnotationTarget.ANNOTATION_CLASS)\n */\npublic annotation class MustBeDocumented\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n\n@JsName("arrayIterator")\ninternal fun arrayIterator(array: dynamic, type: String?) = when (type) {\n    null\n    -> {\n        val arr: Array<dynamic> = array\n        object : Iterator<dynamic> {\n            var index = 0\n            override fun hasNext() = index < arr.size\n            override fun next() = if (index < arr.size) arr[index++] else throw\n                NoSuchElementException("$index")\n        }\n        if ("BooleanArray" == type) booleanArrayIterator(array)\n        if ("ByteArray" == type) byteArrayIterator(array)\n        if ("ShortArray" == type) shortArrayIterator(array)\n        if ("CharArray" == type) charArrayIterator(array)\n        if ("IntArray" == type) intArrayIterator(array)\n        if ("LongArray" == type) longArrayIterator(array)\n        if ("FloatArray" == type) floatArrayIterator(array)\n        if ("DoubleArray" == type) doubleArrayIterator(array)\n        else -> throw\n            IllegalStateException("Unsupported type argument for arrayIterator:\n                $type")\n    }\n}\n\n@JsName("booleanArrayIterator")\ninternal fun booleanArrayIterator(array: BooleanArray) =\n    object : BooleanIterator() {\n        var index = 0\n        override fun hasNext() = index < array.size\n        override fun nextBoolean() = if (index < array.size) array[index++] else throw\n            NoSuchElementException("$index")\n    }\n\n@JsName("byteArrayIterator")\ninternal fun byteArrayIterator(array: ByteArray) =\n    object : ByteIterator() {\n        var index = 0\n        override fun hasNext() = index < array.size\n        override fun nextByte() = if (index < array.size) array[index++] else throw\n            NoSuchElementException("$index")\n    }\n\n@JsName("shortArrayIterator")\ninternal fun\nshortArrayIterator(array: ShortArray) =\n    object : ShortIterator() {\n        var index = 0\n        override fun hasNext() =

```

```

index < array.size\n    override fun nextShort() = if (index < array.size) array[index++] else throw
NoSuchElementException("\$index")\n}\n\n@JsName("charArrayIterator")\ninternal fun charArrayIterator(array:
CharArray) = object : CharIterator() {\n    var index = 0\n    override fun hasNext() = index < array.size\n    override
fun nextChar() = if (index < array.size) array[index++] else throw
NoSuchElementException("\$index")\n}\n\n@JsName("intArrayIterator")\ninternal fun intArrayIterator(array:
IntArray) = object : IntIterator() {\n    var index = 0\n    override fun hasNext() = index < array.size\n    override fun
nextInt() = if (index < array.size) array[index++] else throw
NoSuchElementException("\$index")\n}\n\n@JsName("floatArrayIterator")\ninternal fun
floatArrayIterator(array: FloatArray) = object : FloatIterator() {\n    var index = 0\n    override fun hasNext() = index
< array.size\n    override fun nextFloat() = if (index < array.size) array[index++] else throw
NoSuchElementException("\$index")\n}\n\n@JsName("doubleArrayIterator")\ninternal fun
doubleArrayIterator(array: DoubleArray) = object : DoubleIterator() {\n    var index = 0\n    override fun hasNext()
= index < array.size\n    override fun nextDouble() = if (index < array.size) array[index++] else throw
NoSuchElementException("\$index")\n}\n\n@JsName("longArrayIterator")\ninternal fun longArrayIterator(array:
LongArray) = object : LongIterator() {\n    var index = 0\n    override fun hasNext() = index < array.size\n
override fun nextLong() = if (index < array.size) array[index++] else throw
NoSuchElementException("\$index")\n}\n\n@JsName("PropertyMetadata")\ninternal class
PropertyMetadata(@JsName("callableName") val name:
String)\n\n@JsName("noWhenBranchMatched")\ninternal fun noWhenBranchMatched(): Nothing = throw
NoWhenBranchMatchedException()\n\n@JsName("subSequence")\ninternal fun subSequence(c: CharSequence,
startIndex: Int, endIndex: Int): CharSequence {\n    if (c is String) {\n        return c.substring(startIndex, endIndex)\n
    } else {\n        return c.asDynamic().`subSequence_vux9f0$`(startIndex, endIndex)\n
    }\n}\n\n@JsName("captureStack")\ninternal fun captureStack(@Suppress("UNUSED_PARAMETER")
baseClass: JsClass<in Throwable>, instance: Throwable) {\n    if (js("Error").captureStackTrace) {\n        // Using
uncropped stack traces due to KT-37563.\n        // Precise stack traces are implemented in JS IR compiler and
stdlib\n        js("Error").captureStackTrace(instance);\n    } else {\n        instance.asDynamic().stack = js("new
Error()").stack;\n    }\n}\n\n@JsName("newThrowable")\ninternal fun newThrowable(message: String?, cause:
Throwable?): Throwable {\n    val throwable = js("new Error()")\n    throwable.message = if (jsTypeOf(message)
== "undefined") {\n        if (cause != null) cause.toString() else null\n    } else {\n        message\n    }\n
throwable.cause = cause\n    throwable.name = "Throwable"\n    return
throwable\n}\n\n@JsName("BoxedChar")\ninternal class BoxedChar(val c: Int) : Comparable<Int> {\n    override
fun equals(other: Any?): Boolean {\n        return other is BoxedChar && c == other.c\n    }\n\n    override fun
hashCode(): Int {\n        return c\n    }\n\n    override fun toString(): String {\n        return
js("this.c").unsafeCast<Char>().toString()\n    }\n\n    override fun compareTo(other: Int): Int {\n        return
js("this.c - other").unsafeCast<Int>()\n    }\n\n    @JsName("valueOf")\n    public fun valueOf(): Int {\n
return c\n    }\n}\n\n@kotlin.internal.InlineOnly\ninternal inline fun <T> concat(args: Array<T>): T {\n    val typed
= js("Array")(args.size)\n    for (i in args.indices) {\n        val arr = args[i]\n        if (arr !is Array<*>) {\n
typed[i] = js("[]").slice.call(arr)\n        } else {\n            typed[i] = arr\n        }\n    }\n    return
js("[]").concat.apply(js("[]"), typed);\n}\n\n/** Concat regular Array's and TypedArray's into an Array.\n
*\n@PublishedApi\n@JsName("arrayConcat")\n@Suppress("UNUSED_PARAMETER")\ninternal fun <T>
arrayConcat(a: T, b: T): T {\n    return concat(js("arguments"))\n}\n\n/** Concat primitive arrays. Main use:
prepare vararg arguments.\n
* For compatibility with 1.1.0 the arguments may be a mixture of Array's and
TypedArray's.\n
* If the first argument is TypedArray (Byte-, Short-, Char-, Int-, Float-, and DoubleArray)
returns a TypedArray, otherwise an Array.\n
* If the first argument has the $type$ property (Boolean-, Char-, and
LongArray) copy its value to result.$type$.\n
* If the first argument is a regular Array without the $type$ property
default to arrayConcat.\n
*\n@PublishedApi\n@JsName("primitiveArrayConcat")\n@Suppress("UNUSED_PARAMETER")\ninternal
fun <T> primitiveArrayConcat(a: T, b: T): T {\n    val args: Array<T> = js("arguments")\n    if (a is Array<*> &&

```

```

a.asDynamic().`$type$` === undefined) {\n    return concat(args)\n  } else {\n    var size = 0\n    for (i in
args.indices) {\n      size += args[i].asDynamic().length as Int\n    }\n    val result = js("new
a.constructor(size)")\n    kotlin.copyArrayType(a, result)\n    size = 0\n    for (i in args.indices) {\n      val
arr = args[i].asDynamic()\n      for (j in 0 until arr.length) {\n        result[size++] = arr[j]\n      }\n    }\n
return result\n  }\n}\n\n@JsName("booleanArrayOf")\ninternal fun booleanArrayOf() =
withType("BooleanArray", js("[].slice.call(arguments)"))\n\n@JsName("charArrayOf")\ninternal fun
charArrayOf() = withType("CharArray", js("new
Uint16Array(arguments)"))\n\n@JsName("longArrayOf")\ninternal fun longArrayOf() =
withType("LongArray",
js("[].slice.call(arguments)"))\n\n@JsName("withType")\n\n@kotlin.internal.InlineOnly\n\ninternal inline fun
withType(type: String, array: dynamic): dynamic {\n  array.`$type$` = type\n  return array\n}, /*\n * Copyright
2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\n/**\n *
Function corresponding to JavaScript's `typeof` operator\n
*\n\n@kotlin.internal.InlineOnly\n\n@Suppress("UNUSED_PARAMETER")\n\npublic inline fun jsTypeOf(a: Any?):
String = js("typeof a")\n}, /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\n@file:Suppress("UNUSED_PARAMETER",
"NOTHING_TO_INLINE")\n\npackage kotlin\n\n/**\n * Returns an empty array of the specified type [T].\n
*\n\npublic inline fun <T> emptyArray(): Array<T> = js("[]")\n\n@library\n\npublic fun <T> arrayOf(vararg
elements: T): Array<T> = definedExternally\n\n@library\n\npublic fun doubleArrayOf(vararg elements: Double):
DoubleArray = definedExternally\n\n@library\n\npublic fun floatArrayOf(vararg elements: Float): FloatArray =
definedExternally\n\n@library\n\npublic fun longArrayOf(vararg elements: Long): LongArray =
definedExternally\n\n@library\n\npublic fun intArrayOf(vararg elements: Int): IntArray =
definedExternally\n\n@library\n\npublic fun charArrayOf(vararg elements: Char): CharArray =
definedExternally\n\n@library\n\npublic fun shortArrayOf(vararg elements: Short): ShortArray =
definedExternally\n\n@library\n\npublic fun byteArrayOf(vararg elements: Byte): ByteArray =
definedExternally\n\n@library\n\npublic fun booleanArrayOf(vararg elements: Boolean): BooleanArray =
definedExternally\n\n/**\n * Creates a new instance of the [Lazy] that uses the specified initialization function
[initializer].\n *\n\npublic actual fun <T> lazy(initializer: () -> T): Lazy<T> = UnsafeLazyImpl(initializer)\n\n/**\n *
Creates a new instance of the [Lazy] that uses the specified initialization function [initializer].\n *\n
*\n\nThe [mode] parameter is ignored.\n *\n\npublic actual fun <T> lazy(mode: LazyThreadSafetyMode, initializer: () -> T): Lazy<T> =
UnsafeLazyImpl(initializer)\n\n/**\n * Creates a new instance of the [Lazy] that uses the specified initialization
function [initializer].\n *\n
*\n\nThe [lock] parameter is ignored.\n *\n\npublic actual fun <T> lazy(lock: Any?,
initializer: () -> T): Lazy<T> = UnsafeLazyImpl(initializer)\n\n\n\ninternal fun fillFrom(src: dynamic, dst: dynamic):
dynamic {\n  val srcLen: Int = src.length\n  val dstLen: Int = dst.length\n  var index: Int = 0\n  while (index <
srcLen && index < dstLen) dst[index] = src[index++]\n  return dst\n}\n\n\ninternal fun arrayCopyResize(source:
dynamic, newSize: Int, defaultValue: Any?): dynamic {\n  val result = source.slice(0, newSize)\n
copyArrayType(source, result)\n  var index: Int = source.length\n  if (newSize > index) {\n    result.length =
newSize\n    while (index < newSize) result[index++] = defaultValue\n  }\n  return result\n}\n\n\ninternal fun
<T> arrayPlusCollection(array: dynamic, collection: Collection<T>): dynamic {\n  val result = array.slice()\n
result.length += collection.size\n  copyArrayType(array, result)\n  var index: Int = array.length\n  for (element in
collection) result[index++] = element\n  return result\n}\n\n\ninternal fun <T> fillFromCollection(dst: dynamic,
startIndex: Int, collection: Collection<T>): dynamic {\n  var index = startIndex\n  for (element in collection)
dst[index++] = element\n  return dst\n}\n\n\ninternal inline fun copyArrayType(from: dynamic, to: dynamic) {\n  if
(from.`$type$` !== undefined) {\n    to.`$type$` = from.`$type$`\n  }\n}\n\n\ninternal inline fun jsIsType(obj:
dynamic, jsClass: dynamic) = js("Kotlin").isType(obj, jsClass), /*\n * Copyright 2010-2021 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that

```

```

can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\n/**\n * Creates a Char with the specified
[code].\n *\n * @sample samples.text.Chars.charFromCode\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
actual inline fun Char(code: UShort): Char {\n    return code.toInt().toChar()\n}\n\n",/*\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.coroutines\n\nimport
kotlin.coroutines.intrinsics.COROUTINE_SUSPENDED\n\n@SinceKotlin("1.3")\n@JsName("CoroutineImpl")\n\ninternal abstract class CoroutineImpl(private val resultContinuation: Continuation<Any?>) : Continuation<Any?>
{\n    protected var state = 0\n    protected var exceptionState = 0\n    protected var result: Any? = null\n    protected
var exception: Throwable? = null\n    protected var finallyPath: Array<Int>? = null\n\n    public override val context:
CoroutineContext = resultContinuation.context\n\n    private var intercepted_: Continuation<Any?>? = null\n\n    public fun intercepted(): Continuation<Any?> =\n        intercepted_\n        ?:\n    (context[ContinuationInterceptor]?.interceptContinuation(this) ?: this)\n        .also { intercepted_ = it }\n\n    override fun resumeWith(result: Result<Any?>) {\n        var current = this\n        var currentResult: Any? =
result.getOrNull()\n        var currentException: Throwable? = result.exceptionOrNull()\n        // This loop unrolls
recursion in current.resumeWith(param) to make saner and shorter stack traces on resume\n        while (true) {\n
            with(current) {\n                val completion = resultContinuation\n                // Set result and exception fields in
the current continuation\n                if (currentException == null) {\n                    this.result = currentResult\n
                } else {\n                    state = exceptionState\n                    exception = currentException\n                }\n
            }\n            try {\n                val outcome = doResume()\n                if (outcome === COROUTINE_SUSPENDED)\n                    return\n                currentResult = outcome\n                currentException = null\n            } catch (exception:
dynamic) { // Catch all exceptions\n                currentResult = null\n                currentException =
exception.unsafeCast<Throwable>()\n            }\n            releaseIntercepted() // this state machine instance is
terminating\n\n            if (completion is CoroutineImpl) {\n                // unrolling recursion via loop\n                current = completion\n            } else {\n                // top-level completion reached -- invoke and return\n                currentException?.let {\n                    completion.resumeWithException(it)\n                } ?:\n                completion.resume(currentResult)\n                return\n            }\n        }\n        }\n        }\n        }\n        }\n        private fun
releaseIntercepted() {\n            val intercepted = intercepted_\n            if (intercepted != null && intercepted !== this) {\n
                context[ContinuationInterceptor]!!.releaseInterceptedContinuation(intercepted)\n            }\n            this.intercepted_
= CompletedContinuation // just in case\n        }\n        protected abstract fun doResume(): Any?\n    }\n\n    internal object
CompletedContinuation : Continuation<Any?> {\n        override val context: CoroutineContext\n            get() =
error("This continuation is already complete")\n        override fun resumeWith(result: Result<Any?>) {\n            error("This continuation is already complete")\n        }\n        override fun toString(): String = "This continuation is
already complete"\n    }\n\n",/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\n@file:Suppress("UNCHECKED_CAST",
"RedundantVisibilityModifier")\n\npackage kotlin\n\nimport kotlin.contracts.*\nimport
kotlin.internal.InlineOnly\nimport kotlin.jvm.JvmField\nimport kotlin.jvm.JvmInline\nimport
kotlin.jvm.JvmName\n\n/**\n * A discriminated union that encapsulates a successful outcome with a value of type
[T]\n * or a failure with an arbitrary [Throwable] exception.\n */\n@SinceKotlin("1.3")\n@JvmInline\npublic
value class Result<out T> @PublishedApi internal constructor(\n    @PublishedApi\n    internal val value: Any?\n) :
Serializable {\n    // discovery\n    /**\n     * Returns `true` if this instance represents a successful outcome.\n     *
In this case [isFailure] returns `false`.\n     */\n    public val isSuccess: Boolean\n        get() = value !is Failure\n    /**\n     * Returns `true` if this instance represents a failed outcome.\n     *
In this case [isSuccess] returns `false`.\n     */\n    public val isFailure: Boolean\n        get() = value is Failure\n\n    // value & exception retrieval\n    /**\n     * Returns the
encapsulated value if this instance represents [success][Result.isSuccess] or `null`\n     * if it is
[failure][Result.isFailure].\n     */\n     * This function is a shorthand for `getOrNull { null }` (see [getOrNull]) or\n
     * `fold(onSuccess = { it }, onFailure = { null })` (see [fold]).\n     */\n    @InlineOnly\n    public inline fun

```



```

getOrNull(): T? =\n    when {\n        isFailure -> null\n        else -> value as T\n    }\n    /**\n     *
Returns the encapsulated [Throwable] exception if this instance represents [failure][isFailure] or `null` if it is
[success][isSuccess].\n     *\n     * This function is a shorthand for `fold(onSuccess = { null }, onFailure = { it })`
(see [fold]).\n     */\n    public fun exceptionOrNull(): Throwable? =\n        when (value) {\n            is Failure ->
value.exception\n            else -> null\n        }\n    /**\n     * Returns a string `Success(v)` if this instance represents
[success][Result.isSuccess]\n     * where `v` is a string representation of the value or a string `Failure(x)` if\n     * it
is [failure][isFailure] where `x` is a string representation of the exception.\n     */\n    public override fun toString():
String =\n        when (value) {\n            is Failure -> value.toString() // "Failure($exception)"
            else ->
"Success($value)"\n        }\n    // companion with constructors\n    /**\n     * Companion object for [Result]
class that contains its constructor functions\n     * [success] and [failure].\n     */\n    public companion object {\n
/**\n     * Returns an instance that encapsulates the given [value] as successful value.\n     */\n
@Suppress("INAPPLICABLE_JVM_NAME")\n        @InlineOnly\n        @JvmName("success")\n        public
inline fun <T> success(value: T): Result<T> =\n            Result(value)\n    /**\n     * Returns an instance that
encapsulates the given [Throwable] [exception] as failure.\n     */\n
@Suppress("INAPPLICABLE_JVM_NAME")\n        @InlineOnly\n        @JvmName("failure")\n        public
inline fun <T> failure(exception: Throwable): Result<T> =\n            Result(createFailure(exception))\n    }\n
internal class Failure(\n        @JvmField\n        val exception: Throwable\n    ): Serializable {\n        override fun
equals(other: Any?): Boolean = other is Failure && exception == other.exception\n        override fun hashCode():
Int = exception.hashCode()\n        override fun toString(): String = "Failure($exception)"\n    }\n    /**\n     *
Creates an instance of internal marker [Result.Failure] class to\n     * make sure that this class is not exposed in ABI.\n
*/\n    @PublishedApi\n    @SinceKotlin("1.3")\n    internal fun createFailure(exception: Throwable): Any =\n        Result.Failure(exception)\n    /**\n     * Throws exception if the result is failure. This internal function minimizes\n
*/\n    inlined bytecode for [getOrThrow] and makes sure that in the future we can\n     * add some exception-augmenting
logic here (if needed).\n     */\n    @PublishedApi\n    @SinceKotlin("1.3")\n    internal fun Result<*>.throwOnFailure() {\n
if (value is Result.Failure) throw value.exception\n    }\n    /**\n     * Calls the specified function [block] and returns its
encapsulated result if invocation was successful,\n     * catching any [Throwable] exception that was thrown from the
[block] function execution and encapsulating it as a failure.\n     */\n    @InlineOnly\n    @SinceKotlin("1.3")\n    public
inline fun <R> runCatching(block: () -> R): Result<R> {\n        return try {\n            Result.success(block())\n        } catch
(e: Throwable) {\n            Result.failure(e)\n        }\n    }\n    /**\n     * Calls the specified function [block] with `this` value as
its receiver and returns its encapsulated result if invocation was successful,\n     * catching any [Throwable] exception
that was thrown from the [block] function execution and encapsulating it as a failure.\n     */\n
*/\n    @InlineOnly\n    @SinceKotlin("1.3")\n    public inline fun <T, R> T.runCatching(block: T.() -> R): Result<R> {\n
return try {\n        Result.success(block())\n    } catch (e: Throwable) {\n        Result.failure(e)\n    }\n    }\n    // --
extensions ---\n    /**\n     * Returns the encapsulated value if this instance represents [success][Result.isSuccess] or
throws the encapsulated [Throwable] exception\n     * if it is [failure][Result.isFailure].\n     */\n    *\n     * This function is a
shorthand for `getOrElse { throw it }` (see [getOrElse]).\n     */\n    @InlineOnly\n    @SinceKotlin("1.3")\n    public inline
fun <T> Result<T>.getOrThrow(): T {\n        throwOnFailure()\n        return value as T\n    }\n    /**\n     * Returns the
encapsulated value if this instance represents [success][Result.isSuccess] or the\n     * result of [onFailure] function for
the encapsulated [Throwable] exception if it is [failure][Result.isFailure].\n     */\n    *\n     * Note, that this function rethrows
any [Throwable] exception thrown by [onFailure] function.\n     */\n    *\n     * This function is a shorthand for `fold(onSuccess
= { it }, onFailure = onFailure)` (see [fold]).\n     */\n    @InlineOnly\n    @SinceKotlin("1.3")\n    public inline fun <R, T :
R> Result<T>.getOrElse(onFailure: (exception: Throwable) -> R): R {\n        contract {\n            callsInPlace(onFailure,
InvocationKind.AT_MOST_ONCE)\n        }\n        return when (val exception = exceptionOrNull()) {\n            null ->
value as T\n            else -> onFailure(exception)\n        }\n    }\n    /**\n     * Returns the encapsulated value if this instance
represents [success][Result.isSuccess] or the\n     * [defaultValue] if it is [failure][Result.isFailure].\n     */\n    *\n     * This
function is a shorthand for `getOrElse { defaultValue }` (see [getOrElse]).\n     */\n
*/\n    @InlineOnly\n    @SinceKotlin("1.3")\n    public inline fun <R, T : R> Result<T>.getOrElse(defaultValue: R):
R {\n        if (isFailure) return defaultValue\n        return value as T\n    }\n    /**\n     * Returns the result of [onSuccess] for the

```

encapsulated value if this instance represents [success][Result.isSuccess]\n * or the result of [onFailure] function for the encapsulated [Throwable] exception if it is [failure][Result.isFailure].\n * Note, that this function rethrows any [Throwable] exception thrown by [onSuccess] or by [onFailure] function.\n

```
*\n@InlineOnly\n@SinceKotlin("1.3")\npublic inline fun <R, T> Result<T>.fold(\n    onSuccess: (value: T) -> R,\n    onFailure: (exception: Throwable) -> R): R {\n    contract {\n        callsInPlace(onSuccess,\n            InvocationKind.AT_MOST_ONCE)\n        callsInPlace(onFailure, InvocationKind.AT_MOST_ONCE)\n    }\n    return when (val exception = exceptionOrNull()) {\n        null -> onSuccess(value as T)\n        else -> onFailure(exception)\n    }\n}\n// transformation\n/**\n * Returns the encapsulated result of the given
```

[transform] function applied to the encapsulated value\n * if this instance represents [success][Result.isSuccess] or the\n * original encapsulated [Throwable] exception if it is [failure][Result.isFailure].\n * Note, that this function rethrows any [Throwable] exception thrown by [transform] function.\n * See [mapCatching] for an alternative that encapsulates exceptions.\n *\n@InlineOnly\n@SinceKotlin("1.3")\npublic inline fun <R, T>

```
Result<T>.map(transform: (value: T) -> R): Result<R> {\n    contract {\n        callsInPlace(transform,\n            InvocationKind.AT_MOST_ONCE)\n    }\n    return when {\n        isSuccess -> Result.success(transform(value as T))\n        else -> Result(value)\n    }\n}\n/**\n * Returns the encapsulated result of the given [transform] function applied to the encapsulated value\n * if this instance represents [success][Result.isSuccess] or the\n * original encapsulated [Throwable] exception if it is [failure][Result.isFailure].\n * This function catches any [Throwable] exception thrown by [transform] function and encapsulates it as a failure.\n * See [map] for an alternative that
```

```
rethrows exceptions from `transform` function.\n *\n@InlineOnly\n@SinceKotlin("1.3")\npublic inline fun <R, T> Result<T>.mapCatching(transform: (value: T) -> R): Result<R> {\n    return when {\n        isSuccess -> runCatching { transform(value as T) }\n        else -> Result(value)\n    }\n}\n/**\n * Returns the encapsulated result of the given [transform] function applied to the encapsulated [Throwable] exception\n * if this instance represents [failure][Result.isFailure] or the\n * original encapsulated value if it is [success][Result.isSuccess].\n * Note, that this function rethrows any [Throwable] exception thrown by [transform] function.\n * See [recoverCatching] for an alternative that encapsulates exceptions.\n
```

```
*\n@InlineOnly\n@SinceKotlin("1.3")\npublic inline fun <R, T : R> Result<T>.recover(transform: (exception: Throwable) -> R): Result<R> {\n    contract {\n        callsInPlace(transform, InvocationKind.AT_MOST_ONCE)\n    }\n    return when (val exception = exceptionOrNull()) {\n        null -> this\n        else ->
```

```
Result.success(transform(exception))\n    }\n}\n/**\n * Returns the encapsulated result of the given [transform] function applied to the encapsulated [Throwable] exception\n * if this instance represents [failure][Result.isFailure] or the\n * original encapsulated value if it is [success][Result.isSuccess].\n * This function catches any [Throwable] exception thrown by [transform] function and encapsulates it as a failure.\n * See [recover] for an alternative that rethrows exceptions.\n *\n@InlineOnly\n@SinceKotlin("1.3")\npublic inline fun <R, T : R>
```

```
Result<T>.recoverCatching(transform: (exception: Throwable) -> R): Result<R> {\n    return when (val exception = exceptionOrNull()) {\n        null -> this\n        else -> runCatching { transform(exception) }\n    }\n}\n// "peek" onto value/exception and pipe\n/**\n * Performs the given [action] on the encapsulated [Throwable] exception if this instance represents [failure][Result.isFailure].\n * Returns the original `Result` unchanged.\n
```

```
*\n@InlineOnly\n@SinceKotlin("1.3")\npublic inline fun <T> Result<T>.onFailure(action: (exception: Throwable) -> Unit): Result<T> {\n    contract {\n        callsInPlace(action, InvocationKind.AT_MOST_ONCE)\n    }\n    exceptionOrNull()?.let { action(it) }\n    return this\n}\n/**\n * Performs the given [action] on the encapsulated value if this instance represents [success][Result.isSuccess].\n * Returns the original `Result` unchanged.\n *\n@InlineOnly\n@SinceKotlin("1.3")\npublic inline fun <T> Result<T>.onSuccess(action: (value: T) -> Unit): Result<T> {\n    contract {\n        callsInPlace(action, InvocationKind.AT_MOST_ONCE)\n    }\n    if (isSuccess) action(value as T)\n    return this\n}\n// -----\n"/\n * Copyright 2010-2020 JetBrains
```

s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\npackage kotlin.coroutines\nimport

```
kotlin.contracts.*\nimport kotlin.coroutines.intrinsics.*\nimport kotlin.internal.InlineOnly\n/**\n * Interface representing a continuation after a suspension point that returns a value of type `T`.\n
```

```

*^@SinceKotlin("1.3")\npublic interface Continuation<in T> {\n    /**\n     * The context of the coroutine that
corresponds to this continuation.\n    */\n    public val context: CoroutineContext\n    /**\n     * Resumes the
execution of the corresponding coroutine passing a successful or failed [result] as the\n    * return value of the last
suspension point.\n    */\n    public fun resumeWith(result: Result<T>)\n}\n\n/**\n * Classes and interfaces marked
with this annotation are restricted when used as receivers for extension\n * `suspend` functions. These `suspend`
extensions can only invoke other member or extension `suspend` functions on this particular\n * receiver and are
restricted from calling arbitrary suspension functions.\n
*^@SinceKotlin("1.3")\n@Target(AnnotationTarget.CLASS)\n@Retention(AnnotationRetention.BINARY)\npu
blic annotation class RestrictsSuspension\n\n/**\n * Resumes the execution of the corresponding coroutine passing
[value] as the return value of the last suspension point.\n *^@SinceKotlin("1.3")\n@InlineOnly\npublic inline
fun <T> Continuation<T>.resume(value: T): Unit =\n    resumeWith(Result.success(value))\n\n/**\n * Resumes the
execution of the corresponding coroutine so that the [exception] is re-thrown right after the\n * last suspension
point.\n *^@SinceKotlin("1.3")\n@InlineOnly\npublic inline fun <T>
Continuation<T>.resumeWithException(exception: Throwable): Unit =\n
resumeWith(Result.failure(exception))\n\n\n/**\n * Creates a [Continuation] instance with the given [context] and
implementation of [resumeWith] method.\n *^@SinceKotlin("1.3")\n@InlineOnly\npublic inline fun <T>
Continuation(\n    context: CoroutineContext,\n    crossinline resumeWith: (Result<T>) -> Unit\n): Continuation<T>
=\n    object : Continuation<T> {\n        override val context: CoroutineContext\n            get() = context\n\n
override fun resumeWith(result: Result<T>) =\n                resumeWith(result)\n    }\n\n\n/**\n * Creates a coroutine
without a receiver and with result type [T].\n * This function creates a new, fresh instance of suspendable
computation every time it is invoked.\n * To start executing the created coroutine, invoke `resume(Unit)` on the
returned [Continuation] instance.\n * The [completion] continuation is invoked when the coroutine completes with a
result or an exception.\n * Subsequent invocation of any resume function on the resulting continuation will produce
an [IllegalStateException].\n *^@SinceKotlin("1.3")\n@Suppress("UNCHECKED_CAST")\npublic fun <T>
(suspend () -> T).createCoroutine(\n    completion: Continuation<T>)\n): Continuation<Unit> =\n
SafeContinuation(createCoroutineUnintercepted(completion).intercepted(), COROUTINE_SUSPENDED)\n\n\n/**\n * Creates a coroutine with receiver type [R] and result type [T].\n * This function creates a new, fresh instance of
suspendable computation every time it is invoked.\n * To start executing the created coroutine, invoke
`resume(Unit)` on the returned [Continuation] instance.\n * The [completion] continuation is invoked when the
coroutine completes with a result or an exception.\n * Subsequent invocation of any resume function on the resulting
continuation will produce an [IllegalStateException].\n
*^@SinceKotlin("1.3")\n@Suppress("UNCHECKED_CAST")\npublic fun <R, T> (suspend R.() ->
T).createCoroutine(\n    receiver: R,\n    completion: Continuation<T>)\n): Continuation<Unit> =\n
SafeContinuation(createCoroutineUnintercepted(receiver, completion).intercepted(),
COROUTINE_SUSPENDED)\n\n\n/**\n * Starts a coroutine without a receiver and with result type [T].\n * This
function creates and starts a new, fresh instance of suspendable computation every time it is invoked.\n * The
[completion] continuation is invoked when the coroutine completes with a result or an exception.\n
*^@SinceKotlin("1.3")\n@Suppress("UNCHECKED_CAST")\npublic fun <T> (suspend () ->
T).startCoroutine(\n    completion: Continuation<T>)\n) {\n
createCoroutineUnintercepted(completion).intercepted().resume(Unit)\n}\n\n\n/**\n * Starts a coroutine with receiver
type [R] and result type [T].\n * This function creates and starts a new, fresh instance of suspendable computation
every time it is invoked.\n * The [completion] continuation is invoked when the coroutine completes with a result or
an exception.\n *^@SinceKotlin("1.3")\n@Suppress("UNCHECKED_CAST")\npublic fun <R, T> (suspend
R.() -> T).startCoroutine(\n    receiver: R,\n    completion: Continuation<T>)\n) {\n
createCoroutineUnintercepted(receiver, completion).intercepted().resume(Unit)\n}\n\n\n/**\n * Obtains the current
continuation instance inside suspend functions and suspends\n * the currently running coroutine.\n * In this
function both [Continuation.resume] and [Continuation.resumeWithException] can be used either synchronously
in\n * the same stack-frame where the suspension function is run or asynchronously later in the same thread or\n *

```



```

but not executed\n    this.asDynamic()(completion, true)\n    } else {\n
createCoroutineFromSuspendFunction(completion) {\n    this.asDynamic()(completion)\n    }\n }\n\n/*\n
* Creates unintercepted coroutine with receiver type [R] and result type [T].\n * This function creates a new, fresh
instance of suspendable computation every time it is invoked.\n *\n * To start executing the created coroutine,
invoke `resume(Unit)` on the returned [Continuation] instance.\n * The [completion] continuation is invoked when
coroutine completes with result or exception.\n *\n * This function returns unintercepted continuation.\n *
Invocation of `resume(Unit)` starts coroutine immediately in the invoker's call stack without going through the\n
* [ContinuationInterceptor] that might be present in the completion's [CoroutineContext].\n * It is the invoker's
responsibility to ensure that a proper invocation context is established.\n * Note that [completion] of this function
may get invoked in an arbitrary context.\n *\n * [Continuation.intercepted] can be used to acquire the intercepted
continuation.\n * Invocation of `resume(Unit)` on intercepted continuation guarantees that execution of\n * both the
coroutine and [completion] happens in the invocation context established by\n * [ContinuationInterceptor].\n *\n *
Repeated invocation of any resume function on the resulting continuation corrupts the\n * state machine of the
coroutine and may result in arbitrary behaviour or exception.\n *\n * Since Kotlin("1.3")\npublic actual fun <R, T>
(suspend R.() -> T).createCoroutineUnintercepted(\n receiver: R,\n completion: Continuation<T>)\n):
Continuation<Unit> =\n // Kotlin/JS suspend lambdas have an extra parameter `suspended`\n if
(this.asDynamic().length == 3) {\n // When `suspended` is true the continuation is created, but not executed\n
this.asDynamic()(receiver, completion, true)\n } else {\n createCoroutineFromSuspendFunction(completion)
{\n this.asDynamic()(receiver, completion)\n }\n }\n\n/*\n
* Intercepts this continuation with [ContinuationInterceptor].\n *\n * This function shall be used on the immediate result of
[createCoroutineUnintercepted] or [suspendCoroutineUninterceptedOrReturn],\n * in which case it checks for
[ContinuationInterceptor] in the continuation's [context][Continuation.context],\n * invokes
[ContinuationInterceptor.interceptContinuation], caches and returns the result.\n *\n * If this function is invoked on
other [Continuation] instances it returns `this` continuation unchanged.\n *\n * Since Kotlin("1.3")\npublic actual
fun <T> Continuation<T>.intercepted(): Continuation<T> =\n (this as? CoroutineImpl)?.intercepted() ?:
this\n\nprivate inline fun <T> createCoroutineFromSuspendFunction(\n completion: Continuation<T>,\n
crossinline block: () -> Any?)\n): Continuation<Unit> {\n @Suppress("UNCHECKED_CAST")\n return object
: CoroutineImpl(completion as Continuation<Any?>) {\n override fun doResume(): Any? {\n
exception?.let { throw it }\n return block()\n }\n }\n }\n\n/*\n
* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n *\n * package kotlin.js\n// Mirrors signature from JS IR
BE\n// Used for
js.translator/testData/box/number/mulInt32.kt\n@library\n@JsName("imulEmulated")\n@Suppress("UNUSED_PARAMETER")\ninternal fun imul(x: Int, y: Int): Int =
definedExternally\n\n@Suppress("NOTHING_TO_INLINE")\ninternal inline fun isArrayish(o: dynamic) =
js("Kotlin").isArrayish(o)\n\n/*\n
* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n *\n * package kotlin\n\n// NOTE: Do not author your exceptions as they are written in
this file, instead use this template:\n\n*\npublic open class MyException : Exception {\n constructor() : super()\n
constructor(message: String?) : super(message)\n constructor(message: String?, cause: Throwable?) :
super(message, cause)\n constructor(cause: Throwable?) : super(cause)\n}\n\n*\n\n// TODO: remove primary
constructors, make all secondary\nKT-22055\n\n@Suppress("USELESS_ELVIS_RIGHT_IS_NULL")\npublic actual open class Error actual constructor(message: String?, cause: Throwable?) : Throwable(message, cause ?: null)
{\n actual constructor() : this(null, null)\n actual constructor(message: String?) : this(message, null)\n actual
constructor(cause: Throwable?) : this(undefiend,
cause)\n}\n\n*\n\n@Suppress("USELESS_ELVIS_RIGHT_IS_NULL")\npublic actual open class Exception actual
constructor(message: String?, cause: Throwable?) : Throwable(message, cause ?: null) {\n actual constructor() :
this(null, null)\n actual constructor(message: String?) : this(message, null)\n actual constructor(cause:

```

```

Throwable?) : this(undefined, cause)\n\npublic actual open class RuntimeException actual constructor(message:
String?, cause: Throwable?) : Exception(message, cause) {\n  actual constructor() : this(null, null)\n  actual
constructor(message: String?) : this(message, null)\n  actual constructor(cause: Throwable?) : this(undefined,
cause)\n}\n\npublic actual open class IllegalArgumentException actual constructor(message: String?, cause:
Throwable?) : RuntimeException(message, cause) {\n  actual constructor() : this(null, null)\n  actual
constructor(message: String?) : this(message, null)\n  actual constructor(cause: Throwable?) : this(undefined,
cause)\n}\n\npublic actual open class IllegalStateException actual constructor(message: String?, cause: Throwable?)
: RuntimeException(message, cause) {\n  actual constructor() : this(null, null)\n  actual constructor(message:
String?) : this(message, null)\n  actual constructor(cause: Throwable?) : this(undefined, cause)\n}\n\npublic actual
open class IndexOutOfBoundsException actual constructor(message: String?) : RuntimeException(message) {\n
  actual constructor() : this(null)\n}\n\npublic actual open class ConcurrentModificationException actual
constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause) {\n  actual constructor() :
this(null, null)\n  actual constructor(message: String?) : this(message, null)\n  actual constructor(cause:
Throwable?) : this(undefined, cause)\n}\n\npublic actual open class UnsupportedOperationException actual
constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause) {\n  actual constructor() :
this(null, null)\n  actual constructor(message: String?) : this(message, null)\n  actual constructor(cause:
Throwable?) : this(undefined, cause)\n}\n\npublic actual open class NumberFormatException actual
constructor(message: String?) : IllegalArgumentException(message) {\n  actual constructor() :
this(null)\n}\n\npublic actual open class NullPointerException actual constructor(message: String?) :
RuntimeException(message) {\n  actual constructor() : this(null)\n}\n\npublic actual open class
ClassCastException actual constructor(message: String?) : RuntimeException(message) {\n  actual constructor() :
this(null)\n}\n\npublic actual open class AssertionError\n@SinceKotlin("1.4")\nconstructor(message: String?,
cause: Throwable?) : Error(message, cause) {\n  actual constructor() : this(null)\n  constructor(message: String?) :
this(message, null)\n  actual constructor(message: Any?) : this(message.toString(), message as?
Throwable)\n}\n\npublic actual open class NoSuchElementException actual constructor(message: String?) :
RuntimeException(message) {\n  actual constructor() : this(null)\n}\n\n@SinceKotlin("1.3")\npublic actual open
class ArithmeticException actual constructor(message: String?) : RuntimeException(message) {\n  actual
constructor() : this(null)\n}\n\npublic actual open class NoWhenBranchMatchedException actual
constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause) {\n  actual constructor() :
this(null, null)\n  actual constructor(message: String?) : this(message, null)\n  actual constructor(cause:
Throwable?) : this(undefined, cause)\n}\n\npublic actual open class UninitializedPropertyAccessException actual
constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause) {\n  actual constructor() :
this(null, null)\n  actual constructor(message: String?) : this(message, null)\n  actual constructor(cause:
Throwable?) : this(undefined, cause)\n}\n\n"/**\n * Copyright 2010-2019 JetBrains s.r.o. Use of this source code is
governed by the Apache 2.0 license\n * that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:Suppress("UNUSED_PARAMETER")\n\npackage kotlin.js\n\n@kotlin.internal.InlineOnly\n\ninternal
inline fun jsDeleteProperty(obj: Any, property: Any) {\n  js("delete
obj[property]")\n}\n\n@kotlin.internal.InlineOnly\n\ninternal inline fun jsBitwiseOr(lhs: Any?, rhs: Any?): Int =\n
js("lhs | rhs").unsafeCast<Int>()","/**\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.math\n\n/**\n * Returns this value with the sign bit same as of the
[sign] value.\n */\n * If [sign] is `NaN` the sign of the result is undefined.\n */\n\n@SinceKotlin("1.2")\n\npublic actual
fun Double.withSign(sign: Double): Double {\n  val thisSignBit =
js("Kotlin").doubleSignBit(this).unsafeCast<Int>()\n  val newSignBit =
js("Kotlin").doubleSignBit(sign).unsafeCast<Int>()\n  return if (thisSignBit == newSignBit) this else -
this}\n"/**\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\npackage kotlin\n\n/**\n * Returns a bit representation of the specified floating-point value as [Long]\n */

```

according to the IEEE 754 floating-point "double format" bit layout.

```

*\/n@SinceKotlin("1.2")\/n@library("doubleToBits")\/npublic actual fun Double.toBits(): Long =
definedExternally\/n\/n**\/n * Returns a bit representation of the specified floating-point value as [Long]\/n *
according to the IEEE 754 floating-point "double format" bit layout,\/n * preserving `NaN` values exact layout.\/n
*\/n@SinceKotlin("1.2")\/n@library("doubleToRawBits")\/npublic actual fun Double.toRawBits(): Long =
definedExternally\/n\/n**\/n * Returns the [Double] value corresponding to a given bit representation.\/n
*\/n@SinceKotlin("1.2")\/n@kotlin.internal.InlineOnly\/npublic actual inline fun Double.Companion.fromBits(bits:
Long): Double = js("Kotlin").doubleFromBits(bits).unsafeCast<Double>()\/n\/n**\/n * Returns a bit representation
of the specified floating-point value as [Int]\/n * according to the IEEE 754 floating-point "single format" bit
layout.\/n *\/n * Note that in Kotlin/JS [Float] range is wider than "single format" bit layout can represent,\/n * so
some [Float] values may overflow, underflow or lose their accuracy after conversion to bits and back.\/n
*\/n@SinceKotlin("1.2")\/n@library("floatToBits")\/npublic actual fun Float.toBits(): Int =
definedExternally\/n\/n**\/n * Returns a bit representation of the specified floating-point value as [Int]\/n * according
to the IEEE 754 floating-point "single format" bit layout,\/n * preserving `NaN` values exact layout.\/n *\/n * Note
that in Kotlin/JS [Float] range is wider than "single format" bit layout can represent,\/n * so some [Float] values
may overflow, underflow or lose their accuracy after conversion to bits and back.\/n
*\/n@SinceKotlin("1.2")\/n@library("floatToRawBits")\/npublic actual fun Float.toRawBits(): Int =
definedExternally\/n\/n**\/n * Returns the [Float] value corresponding to a given bit representation.\/n
*\/n@SinceKotlin("1.2")\/n@kotlin.internal.InlineOnly\/npublic actual inline fun Float.Companion.fromBits(bits:
Int): Float =
js("Kotlin").floatFromBits(bits).unsafeCast<Float>()\/n\/n@Suppress("NOTHING_TO_INLINE")\/ninternal
inline fun Long(low: Int, high: Int) = js("Kotlin").Long.fromBits(low, high).unsafeCast<Long>()\/ninternal inline
val Long.low: Int get() = this.asDynamic().getLowBits().unsafeCast<Int>()\/ninternal inline val Long.high: Int get()
= this.asDynamic().getHighBits().unsafeCast<Int>()\/n", /*\/n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin
Programming Language contributors.\/n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\/n *\/nimport kotlin.reflect.KClass\/n@PublishedApi\/ninternal fun <T :
Annotation> KClass<*>.findAssociatedObject(@Suppress("UNUSED_PARAMETER") annotationClass:
KClass<T>): Any? {\/n // This API is not supported in js-v1. Return `null` to be source-compatible with js-ir.\/n
return null\/n}\/n", /*\/n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\/n *
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\/n
*\/npackage kotlin.text\/n\/n**\/n * Returns a string representation of this [Long] value in the specified [radix].\/n
*\/n * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.\/n
*\/n@SinceKotlin("1.2")\/npublic actual fun Long.toString(radix: Int): String =
asDynamic().toString(checkRadix(radix))\/n", /*\/n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\/n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\/n *\/npackage
kotlin.js\/n@PublishedApi\/n@Suppress("NOTHING_TO_INLINE")\/n@JsPolyfill("nif (typeof
Array.prototype.fill === 'undefined') {\/n // Polyfill from https://developer.mozilla.org/en-
US/docs/Web/JavaScript/Reference/Global_Objects/Array/fill#Polyfill\/n Object.defineProperty(Array.prototype,
'fill', {\/n value: function (value) {\/n // Steps 1-2.\/n if (this === null) {\/n throw new
TypeError('this is null or not defined');\/n }\/n var O = Object(this);\/n // Steps 3-5.\/n
var len = O.length >>> 0;\/n // Steps 6-7.\/n var start = arguments[1];\/n var relativeStart = start
>> 0;\/n // Step 8.\/n var k = relativeStart < 0 ?\/n Math.max(len + relativeStart, 0) :\/n
Math.min(relativeStart, len);\/n // Steps 9-10.\/n var end = arguments[2];\/n var
relativeEnd = end === undefined ?\/n len : end >> 0;\/n // Step 11.\/n var finalValue
= relativeEnd < 0 ?\/n Math.max(len + relativeEnd, 0) :\/n Math.min(relativeEnd,
len);\/n // Step 12.\/n while (k < finalValue) {\/n O[k] = value;\/n k++;\/n }\/n
// Step 13.\/n return O;\/n }\/n });\/n\/n[Int8Array, Int16Array, Uint16Array, Int32Array,

```



```
BCiBBCiBChiZBCBCiBcGHhChCiBRBxxEYC40Rx8c6RGUm4GRFRFYRQZ44acG4wRYFEFGJYIIGFIYGwc
GmkEmcGFJF18cYxwFGFGRFGFRJFGkkcYkxRm6aFGEgGmmEmEGRYRFgxxYFRFRFRGQGIIFmIFIGIooGF
FGFYJ4EFmoIRFlxRlXRFRFxIRxIFlIRxMFIgXXIoXRomFRIRxIFmGRJFaL86F4mRxmGoRFRFRFRFlRxGIGR
xmGxmGmxRxxGRFIRRJmmFlIGYRmmIRFlIRFRFlIRFxxGFIgmmRoxImxRFRllGmxRJ4aRFGxmIoRFlxRlX
FRFlIRFxxGIImoGmmRxoIxoIGRmmIRxIFmGRJ8FLRxmFFRFlIRIIIRxxFIRlxRxlFRFRFRooGRlooRomRxxFRIR
JLc8aRmoIoGFllIRFRFRlmgmoIoORGRGRxmGFRllGmxRJYL8lGooYFlIRFRFRFRmllIxxGooRGRIRlxFG
RJxIFRGIFlIRFlmGIGxIoORomF8xRxxFlIIFLFRJLcFxmIoRFRFRFxIRFRxxGxxIoOGmmRRIRJxxIoYRFllGG
RaFEGYJYRxlFRFRFRlRlGGlxRFxEGRJRFRFcy84c8mGcJL8G1WlFRFRGIGmmYFGRGRcGc88RYCYRFIIG
GmmIomGFJYFfoGmlFlIGmmFIFIFGFmoIGIomFJIm8cBhRRxxBC4ECFRFRFIRFRFRFRFRFRFRFIRFRFRFRFR
FRGYLRFcRBRCxxUF8YFMF1WRFYKFRFRFRFRGFRGFRFRllRIRGRFmmIGlooGGY44E46FmxRJRLRY44
U44GmmQRJRFEFRFGRFGIFGRFRFxmgmoIoOGmoIoXRxIoGIGRxxcx4YJFRFRFRFRJLRcFmmIomRx4YFoGG
mRomIGIGmxRJRJRyEYRGmmHRGIFmIGmIoOGFRJYCgcRmmIFomGmmIomGmlFJFmoGooGGIRYFIIIG
RYJRFJFEYCRBRBYRGYGIGFGFlIGomGFRCECECEGRGHcCiBCBCRBRBCBCBRBRcXBCBCRCDCDCD
CiiRBj7CbCiiRBj7b7iCiiRxiCBRbCbBxxCiiRBj7bRmQUY9+V9+VYtOQM9eY43X44Z1WY54XYMQRQRER
LZ12ELZ12RERaRGHGHGR88B88BihBhiChhC8hcZBc8BB8CBFCi8cihBZBC8Z8CLKhCkR8cRZcZc88ZcZc85
Z8ZcZc1WcZc1WcZcZcZcRcRLcLcZcZcZcZc1WlCZ1WZ1WZcZ1WZ1WZ1WZcZcZcRcRcRcBRcixBBCiBBihC
CEBhCCchCGhCRY44LCiRRxxCFrkYRGFRFRFRFRFRFRFRFRFRFRFRG9eY49eY44U49e49e1WYeyUY04VY
48cRcRcRcRc4Y48EIK1Wc1W12U2cKGooUE88KqEl4c8RFxxGm7bkkFUF4kEkFRFRFx8cLcFcfRcLcLc
LcLcLcLcFcfRFEFRcRFEYFEYFJRhClmHnnYG4EHCEGFKGYRbEbhCCiBEciBhCk7bhCiBihCiBBcBhCRhiBh
hCCRhiFkkCFIGlIGlIGFfoGmlcGRL88aRFYRIFIGRYJRGFY14FGJFGYFGIRYFRGIFmoIGIGIYxEJRYFmEFJ
FRFGmoImoIGRFGFmIRJRYFEfclOgIFmlGmlFGFlmGFRllEYFomGo4YIkEoGRFRFRFRFRFRFRcbECk7bRCFo
oG4oGRJRFRFRFRFRSFRFRRCRIgfZFRFlxFFbRF2VRFRFRFR6cRGY41WRG40UX1W44V24Y44X33Y44R
44U1WY50Z5R46YFRFRxxQY44a41W54UYJZYB14W7XC15WZ12YYFEFEFRFRFRFlxRIIRxxa65b86axcZc
RQcR"\n    decodedRangeCategory = decodeVarLenBase64(rangeCategory, fromBase64, 1343)\n
}\n}\n\nprivate fun categoryValueFrom(code: Int, ch: Int): Int {\n    return when {\n        code < 0x20 -> code\n        code < 0x400 -> if ((ch and 1) == 1) code shr 5 else code and 0x1f\n        else ->\n            when (ch % 3) {\n                2 -> code shr 10\n                1 -> (code shr 5) and 0x1f\n                else -> code and 0x1f\n            }\n        }\n}\n}\n}\n\n/**\n * Returns the Unicode general category of this character as an Int.\n */\ninternal fun Char.getCategoryValue(): Int\n{\n    val ch = this.code\n    val index = binarySearchRange(Category.decodedRangeStart, ch)\n    val start =\n        Category.decodedRangeStart[index]\n    val code = Category.decodedRangeCategory[index]\n    val value =\n        categoryValueFrom(code, ch - start)\n    return if (value == 17) CharCategory.UNASSIGNED.value else\n        value\n}\n}\n\ninternal fun decodeVarLenBase64(base64: String, fromBase64: IntArray, resultLength: Int): IntArray\n{\n    val result = IntArray(resultLength)\n    var index = 0\n    var int = 0\n    var shift = 0\n    for (char in base64)\n    {\n        val sixBit = fromBase64[char.code]\n        int = int or ((sixBit and 0x1f) shl shift)\n        if (sixBit < 0x20)\n        {\n            result[index++] = int\n            int = 0\n            shift = 0\n        } else {\n            shift += 5\n        }\n    }\n    return result\n}\n"/**\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\n// NOTE: THIS FILE IS AUTO-GENERATED by the\nGenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport\nkotlin.js.*\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Reverses elements in the list in-  
place.\n */\npublic actual fun <T> MutableList<T>.reverse(): Unit {\n    val midPoint = (size / 2) - 1\n    if\n    (midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n"/**\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage\nkotlin.text\n\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt\n// See:  
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n// 37 ranges totally\nprivate object Digit {\n    internal val rangeStart = intArrayOf(\n        0x0030, 0x0660, 0x06f0, 0x07c0, 0x0966, 0x09e6, 0x0a66, 0x0ae6,
```

```

0x0b66, 0x0be6, 0x0c66, 0x0ce6, 0x0d66, 0x0de6, 0x0e50, 0x0ed0, 0x0f20, 0x1040, 0x1090, 0x17e0, \n
0x1810, 0x1946, 0x19d0, 0x1a80, 0x1a90, 0x1b50, 0x1bb0, 0x1c40, 0x1c50, 0xa620, 0xa8d0, 0xa900, 0xa9d0,
0xa9f0, 0xaa50, 0xabf0, 0xff10, \n  )\n}\n\n/**\n * Returns the index of the largest element in [array] smaller or
equal to the specified [needle],\n * or -1 if [needle] is smaller than the smallest element in [array].\n */\ninternal fun
binarySearchRange(array: IntArray, needle: Int): Int {\n    var bottom = 0\n    var top = array.size - 1\n    var middle
= -1\n    var value = 0\n    while (bottom <= top) {\n        middle = (bottom + top) / 2\n        value = array[middle]\n
        if (needle > value)\n            bottom = middle + 1\n        else if (needle == value)\n            return middle\n
        else\n            top = middle - 1\n    }\n    return middle - (if (needle < value) 1 else 0)\n}\n\n/**\n * Returns an integer
from 0..9 indicating the digit this character represents,\n * or -1 if this character is not a digit.\n */\ninternal fun
Char.digitToIntImpl(): Int {\n    val ch = this.code\n    val index = binarySearchRange(Digit.rangeStart, ch)\n    val
diff = ch - Digit.rangeStart[index]\n    return if (diff < 10) diff else -1\n}\n\n/**\n * Returns `true` if this character
is a digit.\n */\ninternal fun Char.isDigitImpl(): Boolean {\n    return digitToIntImpl() >= 0\n},"*\n * Copyright
2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\npackage kotlin.text\n\n/\nNOTE: THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n/\n\n// 222 ranges totally\nprivate object Letter {\n
val decodedRangeStart: IntArray\n    val decodedRangeLength: IntArray\n    val decodedRangeCategory: IntArray\n
    \n    init {\n        val toBase64 =
        \"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/\n        val fromBase64 =
        IntArray(128)\n        for (i in toBase64.indices) {\n            fromBase64[toBase64[i].code] = i\n        }\n
        \n        //
rangeStartDiff.length = 356\n        val rangeStartDiff =
        \"hCgBpCQGYHZH5BRpBPPPPPRMP5BPPICPP6BkEPPPPcPXPzBvBrB3BOiDoBHwD+E3DauCnFmBmB2D
6E1BIBTiBmBIBP5BhBiBrBvBjBqBnBPRtBiCmCtBIB0BmB5BiB7BmBgEmChBzGCoEoGVpBsFrhBPqKQ2B
wBYoFgB4CJuTiEvBuCuDrF5DgEgFIJ1DgFmBQtBsBRGsB+BPiBID1EIjDPRPPPQPPPPPGQSQS/DxENVNU+
B9zCwBwBPPCkDPNnBPqDYY1R8B7FkFgTgwGgwUwmBgKwBuBScmEP/BPPPPPrBP8B7F1B/ErBqC6B7B
iBmBfQsBUwCw/KwqIwLwETPcPjQgJxFgBIBsD\"\n        val diff = decodeVarLenBase64(rangeStartDiff,
fromBase64, 222)\n        val start = IntArray(diff.size)\n        for (i in diff.indices) {\n            if (i == 0) start[i] =
diff[i]\n            else start[i] = start[i - 1] + diff[i]\n        }\n        decodedRangeStart = start\n        \n        //
rangeLength.length = 328\n        val rangeLength =
        \"aaMBXHYH5BRpBPPPPPRMP5BPPICPPzBDOOPPcPXPzBvBjB3BOhDmBBpB7DoDYxB+EiBP1DoExBkB
QhBekBPmBgBhBctBiBMWOOXhCsBpBkBUV3Ba4BkB0D1CgBXgBtD4FSdBfPhBpKP0BvBXjEQ2CGsT8Dh
BtCqDpFvD1D3E0IrD2EkBjRBD0BsB+BPiBIB1EIjDPPPPPPPPPPGPPMNLsBNPNPKCvBvBPPCkDPBmBPh
DXXgD4B6FzEgDguG9vUtkB9JcuBSckEP/BPPPPPPBPf4FrBjEhBpC3B5BKawPrBOWCk/KsCuLqDHPbPxPsFt
EaaqDL\"\n        decodedRangeLength = decodeVarLenBase64(rangeLength, fromBase64, 222)\n        \n        //
rangeCategory.length = 959\n        val rangeCategory =
        \"GFjgggUHGGFFZZZmzpz5qB6s6020B60ptltB6smt2sB60mz22B1+vv+8BZZ5s2850BW5q1ymtB506smzBF3q1
q1qB1q1q1+Bgii4wDTm74g3KiggxqM60q1q1Bq1o1q1BF1qlrqrBZ2q5wprBGFZWWZGHFsjiioLowgmOowjkw
CkgoiIk7ligGogioBkwkiYkzj2oNoi+sbkwj04DghhkQ8wgiYkgoioDsgnkwC4gikQ//v+85BkwvoIsgoyI4ygu0whiw
Eowri4CoghsJowgqYowgm4DkwgsY/nwnzPowhmYkg6wI8yggZswikwHgxgmIoxgqYkkgk4DkxgmIkgoioBsgsso
BgzgyI8g9gL8g9ki0wgwJoxgkoC0wgioFkw/wI0w53iF4gioYowjmgBHGq1qkgwBF1q1q8qBhwghuIwghyKk0go
QkwgoQk3goQHGFHkyg0pBgxj6IoinkxDswno7Ikwhz9Bo0gioB8z48Rwli0xN0mpjoX8w78pDwltoqKHFGGwwg
sIHFH3q1q16BFHWFZ1q10q1B2q1wq1B1q10q1B2q1yq1B6q1gq1Biq1qhxBir1qp1Bqt1q1qB1g1q1+B//3q16B///q
1qBH/qlq9Bholq9B1i00a1q10qD1op1HkwmigEigiy6Cptogq1Bixo1kDq7/j00B2qgoBWGFm1lz50B6s5q1+BG
WhggzhwBFFhgk4//Bo2jigE8wguI8wguI8wguUog1qoB4qjmIwwi2KkgYHHH4IBgiFWkgIWoghssMmz5smrBZ
3q1y50B5sm7gzBtz1smzB5smz50BqzqtzmzB5sgzqzBF2//9//5BowgoIwmnkzPkwgk4C8ys65BkgoqI0wgy6FghquZo
2giY0ghiIsgH24B4ghsQ8QF/v1q1OFs008iCHHF1qggz/B8wg6Iznv+//B08QgohsJK0QGfK7hsQ4gB\"\n
        decodedRangeCategory = decodeVarLenBase64(rangeCategory, fromBase64, 222)\n    }\n}\n\n/**\n * Returns
`true` if this character is a letter.\n */\ninternal fun Char.isLetterImpl(): Boolean {\n    return getLetterType() !=

```

```

0\n}\n\n/**\n * Returns `true` if this character is a lower case letter, or it has contributory property
`Other_Lowercase`.\n */\ninternal fun Char.isLowerCaseImpl(): Boolean {\n    return getLetterType() == 1 ||
code.isOtherLowercase()\n}\n\n/**\n * Returns `true` if this character is an upper case letter, or it has contributory
property `Other_Uppercase`.\n */\ninternal fun Char.isUpperCaseImpl(): Boolean {\n    return getLetterType() == 2
|| code.isOtherUppercase()\n}\n\n/**\n * Returns\n * - `1` if the character is a lower case letter,\n * - `2` if the
character is an upper case letter,\n * - `3` if the character is a letter but not a lower or upper case letter,\n * - `0`
otherwise.\n */\nprivate fun Char.getLetterType(): Int {\n    val ch = this.code\n    val index =
binarySearchRange(Letter.decodedRangeStart, ch)\n    val rangeStart = Letter.decodedRangeStart[index]\n    val
rangeEnd = rangeStart + Letter.decodedRangeLength[index] - 1\n    val code =
Letter.decodedRangeCategory[index]\n    if (ch > rangeEnd) {\n        return 0\n    }\n    val lastTwoBits = code
and 0x3\n    if (lastTwoBits == 0) { // gap pattern\n        var shift = 2\n        var threshold = rangeStart\n        for (i
in 0..1) {\n            threshold += (code shr shift) and 0x7f\n            if (threshold > ch) {\n                return 3\n
            }\n            shift += 7\n            threshold += (code shr shift) and 0x7f\n            if (threshold > ch) {\n                return
0\n            }\n            shift += 7\n        }\n        return 3\n    }\n    if (code <= 0x7) {\n        return lastTwoBits\n
    }\n    val distance = (ch - rangeStart)\n    val shift = if (code <= 0x1F) distance % 2 else distance\n    return (code
shr (2 * shift)) and 0x3\n}\n\n", "/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the
GenerateUnicodeData.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nprivate object
OtherLowercase {\n    internal val otherLowerStart = intArrayOf(\n        0x00aa, 0x00ba, 0x02b0, 0x02c0, 0x02e0,
0x0345, 0x037a, 0x1d2c, 0x1d78, 0x1d9b, 0x2071, 0x207f, 0x2090, 0x2170, 0x24d0, 0x2c7c, 0xa69c, 0xa770,
0xa7f8, 0xab5c, \n    )\n    internal val otherLowerLength = intArrayOf(\n        1, 1, 9, 2, 5, 1, 1, 63, 1, 37, 1, 1, 13,
16, 26, 2, 2, 1, 2, 4, \n    )\n}\n\ninternal fun Int.isOtherLowercase(): Boolean {\n    val index =
binarySearchRange(OtherLowercase.otherLowerStart, this)\n    return index >= 0 && this <
OtherLowercase.otherLowerStart[index] + OtherLowercase.otherLowerLength[index]\n}\n\n", "/*\n * Copyright
2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/\n//
NOTE: THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\ninternal fun Int.isOtherUppercase(): Boolean
{\n    return this in 0x2160..0x216f\n        || this in 0x24b6..0x24cf\n}\n\n", "/*\n * Copyright 2010-2022 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/\n// NOTE: THIS FILE IS
AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.js.*\n\n/**\n * Returns a
character at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this char
sequence.\n * \n * @sample samples.collections.Collections.elements.elementAt\n */\npublic actual fun
CharSequence.elementAt(index: Int): Char {\n    return elementAtOrElse(index) { throw
IndexOutOfBoundsException("index: $index, length: $length") }\n}\n\n", "/*\n * Copyright 2010-2021 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/\n// NOTE: THIS FILE IS
AUTO-GENERATED by the GenerateUnicodeData.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n4 ranges totally\n\ninternal fun
Char.titlecaseCharImpl(): Char {\n    val code = this.code\n    // Letters repeating <Lu, Lt, Ll> sequence and code of
the Lt is a multiple of 3, e.g. <\u01c4, \u01c5, \u01c6>\n    if (code in 0x01c4..0x01cc || code in 0x01f1..0x01f3) {\n
        return (3 * ((code + 1) / 3)).toChar()\n    }\n    // Lower case letters whose title case mapping equivalent is equal
to the original letter\n    if (code in 0x10d0..0x10fa || code in 0x10fd..0x10ff) {\n        return this\n    }\n    return
uppercaseChar()\n}\n\n", "/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the

```

```

license/LICENSE.txt file.\n *\n\npackage kotlin.collections\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED
by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport
kotlin.js.*\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns an element at the given
[index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAt\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic actual fun UIntArray.elementAt(index: Int):
UInt {\n    return elementAtOrElse(index) { throw IndexOutOfBoundsException("\nindex: $index, size: $size")\n
}\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic actual fun ULongArray.elementAt(index: Int):
ULong {\n    return elementAtOrElse(index) { throw IndexOutOfBoundsException("\nindex: $index, size: $size")\n
}\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic actual fun UByteArray.elementAt(index: Int):
UByte {\n    return elementAtOrElse(index) { throw IndexOutOfBoundsException("\nindex: $index, size: $size")\n
}\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic actual fun UShortArray.elementAt(index: Int):
UShort {\n    return elementAtOrElse(index) { throw IndexOutOfBoundsException("\nindex: $index, size: $size")\n
}\n}\n\n/**\n * Returns a [List] that wraps the original array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic actual fun UIntArray.asList(): List<UInt> {\n
return object : AbstractList<UInt>(), RandomAccess {\n    override val size: Int get() = this@asList.size\n
override fun isEmpty(): Boolean = this@asList.isEmpty()\n    override fun contains(element: UInt): Boolean =
this@asList.contains(element)\n    override fun get(index: Int): UInt {\n
AbstractList.checkElementIndex(index, size)\n        return this@asList[index]\n    }\n    override fun
indexOf(element: UInt): Int {\n        @Suppress("USELESS_CAST")\n        if ((element as Any?) !is UInt)
return -1\n        return this@asList.indexOf(element)\n    }\n    override fun lastIndexOf(element: UInt): Int
{\n        @Suppress("USELESS_CAST")\n        if ((element as Any?) !is UInt) return -1\n        return
this@asList.lastIndexOf(element)\n    }\n}\n}\n\n/**\n * Returns a [List] that wraps the original array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic actual fun ULongArray.asList(): List<ULong>
{\n    return object : AbstractList<ULong>(), RandomAccess {\n    override val size: Int get() = this@asList.size\n
override fun isEmpty(): Boolean = this@asList.isEmpty()\n    override fun contains(element: ULong):
Boolean = this@asList.contains(element)\n    override fun get(index: Int): ULong {\n
AbstractList.checkElementIndex(index, size)\n        return this@asList[index]\n    }\n    override fun
indexOf(element: ULong): Int {\n        @Suppress("USELESS_CAST")\n        if ((element as Any?) !is
ULong) return -1\n        return this@asList.indexOf(element)\n    }\n    override fun lastIndexOf(element:
ULong): Int {\n        @Suppress("USELESS_CAST")\n        if ((element as Any?) !is ULong) return -1\n
return this@asList.lastIndexOf(element)\n    }\n}\n}\n\n/**\n * Returns a [List] that wraps the original
array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic actual fun UByteArray.asList():
List<UByte> {\n    return object : AbstractList<UByte>(), RandomAccess {\n    override val size: Int get() =
this@asList.size\n    override fun isEmpty(): Boolean = this@asList.isEmpty()\n    override fun
contains(element: UByte): Boolean = this@asList.contains(element)\n    override fun get(index: Int): UByte {\n
AbstractList.checkElementIndex(index, size)\n        return this@asList[index]\n    }\n    override fun
indexOf(element: UByte): Int {\n        @Suppress("USELESS_CAST")\n        if ((element as Any?) !is
UByte) return -1\n        return this@asList.indexOf(element)\n    }\n    override fun lastIndexOf(element:
UByte): Int {\n        @Suppress("USELESS_CAST")\n        if ((element as Any?) !is UByte) return -1\n
return this@asList.lastIndexOf(element)\n    }\n}\n}\n\n/**\n * Returns a [List] that wraps the original array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic actual fun UShortArray.asList(): List<UShort>

```

```

{\n  return object : AbstractList<UShort>(), RandomAccess {\n      override val size: Int get() = this@asList.size\n      override fun isEmpty(): Boolean = this@asList.isEmpty()\n      override fun contains(element: UShort): Boolean = this@asList.contains(element)\n      override fun get(index: Int): UShort {\n  AbstractList.checkElementIndex(index, size)\n      return this@asList[index]\n  }\n  override fun indexOf(element: UShort): Int {\n      @Suppress("USELESS_CAST")\n      if ((element as Any?) !is UShort) return -1\n      return this@asList.indexOf(element)\n  }\n  override fun lastIndexOf(element: UShort): Int {\n      @Suppress("USELESS_CAST")\n      if ((element as Any?) !is UShort) return -1\n      return this@asList.lastIndexOf(element)\n  }\n  }\n}\n\n"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n/\n\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n/\n\n// 9 ranges totally\n/**\n * Returns `true` if this character is a whitespace.\n */\ninternal fun Char.isWhitespaceImpl(): Boolean {\n  val ch = this.code\n  return ch in 0x0009..0x000d\n      || ch in 0x001c..0x0020\n      || ch == 0x00a0\n      || ch > 0x1000 && (\n  ch == 0x1680\n      || ch in 0x2000..0x200a\n      || ch == 0x2028\n      || ch == 0x2029\n      || ch == 0x202f\n      || ch == 0x205f\n      || ch == 0x3000\n      )\n}\n\n"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\n\npublic actual fun interface Comparator<T> {\n  @JsName("compare")\n  public actual fun compare(a: T, b: T): Int\n}\n\n"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\nimport kotlin.annotation.AnnotationTarget.*\n\n@Target(FUNCTION)\n@Deprecated("Use inline extension function with body using dynamic")\npublic annotation class nativeGetter\n\n@Target(FUNCTION)\n@Deprecated("Use inline extension function with body using dynamic")\npublic annotation class nativeSetter\n\n@Target(FUNCTION)\n@Deprecated("Use inline extension function with body using dynamic")\npublic annotation class nativeInvoke\n\n@Target(CLASS, FUNCTION, PROPERTY)\ninternal annotation class library(public val name: String = "")\n\n@Target(CLASS)\ninternal annotation class marker\n\n/**\n * Gives a declaration (a function, a property or a class) specific name in JavaScript.\n */\n * This may be useful in the following cases:\n * * There are two functions for which the compiler gives same name in JavaScript, you can mark one with `@JsName(...)` to prevent the compiler from reporting error.\n * * You are writing a JavaScript library in Kotlin. The compiler produces mangled names for functions with parameters, which is unnatural for usual JavaScript developer. You can put `@JsName(...)` on functions you want to be available from JavaScript.\n * * For some reason you want to rename declaration, e.g. there's common term in JavaScript for a concept provided by the declaration, which is uncommon in Kotlin.\n */\n * Example:\n * ```\n * kotlin\n * class Person(val name: String) {\n *     fun hello() {\n *         println("Hello $name!")\n *     }\n *     @JsName("helloWithGreeting")\n *     fun hello(greeting: String) {\n *         println("$greeting $name!")\n *     }\n * }\n * @property name the name which compiler uses both for declaration itself and for all references to the declaration.\n *     It's required to denote a valid JavaScript identifier.\n */\n * @Retention(AnnotationRetention.BINARY)\n * @Target(CLASS, FUNCTION, PROPERTY, CONSTRUCTOR, PROPERTY_GETTER, PROPERTY_SETTER)\n * public actual annotation class JsName(actual val name: String)\n\n/**\n * Denotes an `external` declaration that must be imported from native JavaScript library.\n */\n * The compiler produces the code relevant for the target module system, for example, in case of CommonJS, it will import the declaration via the `require(...)` function.\n */\n * The annotation can be used on top-level external declarations (classes, properties, functions) and files.\n */\n * In case of file (which can't be `external`) the following rule applies: all the declarations in the file must be `external`. By applying `@JsModule(...)` on a file you tell the compiler to import a JavaScript object that contain all the declarations from the file.\n */\n * Example:\n * ```\n * kotlin\n * @JsModule("jquery")\n * external abstract class JQuery() {\n *     // some declarations here\n * }\n * @JsModule("jquery")\n * external fun JQuery(element: Element): JQuery\n * ```\n

```

*\n * @property import name of a module to import declaration from.\n * It is not interpreted by the Kotlin compiler, it's passed as is directly to the target module system.\n * \n * @see JsNonModule\n *\n *\n @Retention(AnnotationRetention.BINARY)\n @Target(CLASS, PROPERTY, FUNCTION, FILE)\n public annotation class JsModule(val import: String)\n \n /**\n * Denotes an `external` declaration that can be used without module system.\n * \n * By default, an `external` declaration is available regardless your target module system.\n * \n * However, by applying [JsModule] annotation you can make a declaration unavailable to *plain* module system.\n * \n * Some JavaScript libraries are distributed both as a standalone downloadable piece of JavaScript and as a module available\n * as an npm package.\n * \n * To tell the Kotlin compiler to accept both cases, you can augment [JsModule] with the `@JsNonModule` annotation.\n * \n * For example:\n * \n * ``` kotlin\n * @JsModule("jquery")\n * @JsNonModule\n * @JsName("\$")\n * external abstract class JQuery() {\n * // some declarations here\n * }\n *\n * @JsModule("jquery")\n * @JsNonModule\n * @JsName("\$")\n * external fun JQuery(element: Element): JQuery\n * ```\n * \n * @see JsModule\n *\n *\n @Retention(AnnotationRetention.BINARY)\n @Target(CLASS, PROPERTY, FUNCTION, FILE)\n public annotation class JsNonModule\n \n /**\n * Adds prefix to `external` declarations in a source file.\n * \n * JavaScript does not have concept of packages (namespaces). They are usually emulated by nested objects.\n * \n * The compiler turns references to `external` declarations either to plain unprefix names (in case of *plain* modules)\n * or to plain imports.\n * \n * However, if a JavaScript library provides its declarations in packages, you won't be satisfied with this.\n * \n * You can tell the compiler to generate additional prefix before references to `external` declarations using the `@JsQualifier(...)` annotation.\n * \n * Note that a file marked with the `@JsQualifier(...)` annotation can't contain non-`external` declarations.\n * \n * Example:\n * \n * ```\n * @file:JsQualifier("my.jsPackageName")\n * package some.kotlinPackage\n * \n * external fun foo(x: Int)\n *\n * external fun bar(): String\n * ```\n * \n * @property value the qualifier to add to the declarations in the generated code.\n * \n * It must be a sequence of valid JavaScript identifiers separated by the `.` character.\n * \n * Examples of valid qualifiers are: `foo`, `bar.Baz`, `_.\$.f`.\n * \n * @see JsModule\n *\n *\n @Retention(AnnotationRetention.BINARY)\n @Target(AnnotationTarget.FILE)\n public annotation class JsQualifier(val value: String)\n \n /**\n * Exports top-level declaration on JS platform.\n * \n * Compiled module exposes declarations that are marked with this annotation without name mangling.\n * \n * This annotation can be applied to either files or top-level declarations.\n * \n * It is currently prohibited to export the following kinds of declarations:\n * \n * * `expect` declarations\n * * inline functions with reified type parameters\n * * suspend functions\n * * secondary constructors without `@JsName`\n * * extension properties\n * * enum classes\n * * annotation classes\n * \n * Signatures of exported declarations must only contain `exportable` types:\n * \n * * `dynamic`, `Any`, `String`, `Boolean`, `Byte`, `Short`, `Int`, `Float`, `Double`\n * * `BooleanArray`, `ByteArray`, `ShortArray`, `IntArray`, `FloatArray`, `DoubleArray`\n * * `Array<exportable-type>`\n * * Function types with exportable parameters and return types\n * * `external` or `@JsExport` classes and interfaces\n * * Nullable counterparts of types above\n * * Unit return type. Must not be nullable\n * \n * This annotation is experimental, meaning that restrictions mentioned above are subject to change.\n *\n *\n @ExperimentalJsExport\n @Retention(AnnotationRetention.BINARY)\n @Target(CLASS, PROPERTY, FUNCTION, FILE)\n @SinceKotlin("1.3")\n public actual annotation class JsExport\n \n /**\n * Forces a top-level property to be initialized eagerly, opposed to lazily on the first access to file and/or property.\n *\n *\n @ExperimentalStdlibApi\n @Retention(AnnotationRetention.BINARY)\n @Target(AnnotationTarget.PROPERTY)\n @SinceKotlin("1.6")\n @Deprecated("This annotation is a temporal migration assistance and may be removed in the future releases, please consider filing an issue about the case where it is needed")\n public annotation class EagerInitialization\n \n /**\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * \n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n *\n package kotlin.jvm\n \n // these are used in common generated code in stdlib\n \n // TODO: find how to deprecate these\n *\n *\n @Target(AnnotationTarget.FIELD)\n @Retention(AnnotationRetention.SOURCE)\n public actual annotation class Volatile\n *\n *\n @Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY_GETTER, AnnotationTarget.PROPERTY_SETTER)\n @Retention(AnnotationRetention.SOURCE)\n public actual annotation

```

class Synchronized<E>() {
    /**
     * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
     * contributors. Use of this source code is governed by the Apache 2.0 license that can be found in the
     * license/LICENSE.txt file.
     */
    package kotlin.collections

    /**
     * Provides a skeletal implementation of the [MutableCollection] interface.
     */
    @param E the type of elements contained in the collection. The collection is
    invariant in its element type.
    /**
     * public actual abstract class AbstractMutableCollection<E> protected actual
     * constructor() : AbstractCollection<E>(), MutableCollection<E> {
     *     actual abstract override fun add(element: E):
     *     Boolean
     *     actual override fun remove(element: E): Boolean {
     *         checkIsMutable()
     *         val iterator =
     *         iterator()
     *         while (iterator.hasNext()) {
     *             if (iterator.next() == element) {
     *                 iterator.remove()
     *                 return true
     *             }
     *         }
     *         return false
     *     }
     *     actual override fun addAll(elements:
     *     Collection<E>): Boolean {
     *         checkIsMutable()
     *         var modified = false
     *         for (element in elements) {
     *             if (add(element)) modified = true
     *         }
     *         return modified
     *     }
     *     actual override fun
     *     removeAll(elements: Collection<E>): Boolean {
     *         checkIsMutable()
     *         return (this as
     *         MutableIterable<E>).removeAll { it in elements }
     *     }
     *     actual override fun retainAll(elements:
     *     Collection<E>): Boolean {
     *         checkIsMutable()
     *         return (this as MutableIterable<E>).removeAll { it !in
     *         elements }
     *     }
     *     actual override fun clear(): Unit {
     *         checkIsMutable()
     *         val iterator = this.iterator()
     *         while (iterator.hasNext()) {
     *             iterator.next()
     *             iterator.remove()
     *         }
     *     }
     * }
    @Deprecated("Provided so that subclasses inherit this function", level = DeprecationLevel.HIDDEN)
    @JsName("toJSON")
    protected fun toJSON(): Any = this.toArray()

    /**
     * This method is called
     * every time when a mutating method is called on this mutable collection.
     * Mutable collections that are built
     * (frozen) must throw `UnsupportedOperationException`.
     */
    internal open fun checkIsMutable(): Unit {
    }

    /**
     * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
     * Use of
     * this source code is governed by the Apache 2.0 license that can be found in the
     * license/LICENSE.txt file.
     */
    package kotlin.collections

    /**
     * Based on GWT AbstractList
     * Copyright 2007 Google Inc.
     */
    package kotlin.collections

    /**
     * Provides a skeletal implementation of the [MutableList] interface.
     */
    @param E the type of elements contained in the list. The list is invariant in its element type.
    /**
     * public actual abstract class
     * AbstractMutableList<E> protected actual constructor() : AbstractMutableCollection<E>(),
     * MutableList<E> {
     *     protected var modCount: Int = 0
     *     abstract override fun add(index: Int, element: E): Unit
     *     abstract override
     *     fun removeAt(index: Int): E
     *     abstract override fun set(index: Int, element: E): E
     *     /**
     *      * Adds the
     *      * specified element to the end of this list.
     *      * @return `true` because the list is always modified as the result
     *      * of this operation.
     *      */
     *     actual override fun add(element: E): Boolean {
     *         checkIsMutable()
     *         add(size,
     *         element)
     *         return true
     *     }
     *     actual override fun addAll(index: Int, elements: Collection<E>): Boolean {
     *         AbstractList.checkPositionIndex(index, size)
     *         checkIsMutable()
     *         var _index = index
     *         var
     *         changed = false
     *         for (e in elements) {
     *             add(_index++, e)
     *             changed = true
     *         }
     *         return
     *         changed
     *     }
     *     actual override fun clear() {
     *         checkIsMutable()
     *         removeRange(0, size)
     *     }
     *     actual override fun removeAll(elements: Collection<E>): Boolean {
     *         checkIsMutable()
     *         return
     *         removeAll { it in elements }
     *     }
     *     actual override fun retainAll(elements: Collection<E>): Boolean {
     *         checkIsMutable()
     *         return removeAll { it !in elements }
     *     }
     *     actual override fun iterator():
     *     MutableIterator<E> = IteratorImpl()
     *     actual override fun contains(element: E): Boolean = indexOf(element) >=
     *     0
     *     actual override fun indexOf(element: E): Int {
     *         for (index in 0..lastIndex) {
     *             if (get(index) ==
     *             element) {
     *                 return index
     *             }
     *         }
     *         return -1
     *     }
     *     actual override fun
     *     lastIndexOf(element: E): Int {
     *         for (index in lastIndex downTo 0) {
     *             if (get(index) == element) {
     *                 return index
     *             }
     *         }
     *         return -1
     *     }
     *     actual override fun listIterator():
     *     MutableListIterator<E> = listIterator(0)
     *     actual override fun listIterator(index: Int): MutableListIterator<E> =
     *     ListIteratorImpl(index)
     *     actual override fun subList(fromIndex: Int, toIndex: Int): MutableList<E> =
     *     SubList(this, fromIndex, toIndex)
     *     /**
     *      * Removes the range of elements from this list starting from
     *      * [fromIndex] and ending with but not including [toIndex].
     *      */
     *     protected open fun removeRange(fromIndex:
     *     Int, toIndex: Int) {
     *         val iterator = listIterator(fromIndex)
     *         repeat(toIndex - fromIndex) {
     *             iterator.next()
     *             iterator.remove()
     *         }
     *     }
     *     /**
     *      * Compares this list with another list instance

```

```

with the ordered structural equality.\n
 * \n
 * @return true, if [other] instance is a [List] of the same size, which
contains the same elements in the same order.\n
 * \n
 override fun equals(other: Any?): Boolean {\n
 if (other
=== this) return true\n
 if (other !is List<*>) return false\n
 return AbstractList.orderedEquals(this, other)\n
 }\n
 /**\n
 * Returns the hash code value for this list.\n
 * \n
 override fun hashCode(): Int =
AbstractList.orderedHashCode(this)\n
\n
 private open inner class IteratorImpl : MutableIterator<E> {\n
 /**
the index of the item that will be returned on the next call to [next]()\n
 * \n
 protected var index = 0\n
 /** the
index of the item that was returned on the previous call to [next]()\n
 * \n
 * or [ListIterator.previous]() (for
`ListIterator`),\n
 * -1 if no such item exists\n
 * \n
 protected var last = -1\n
 override fun
hasNext(): Boolean = index < size\n
 override fun next(): E {\n
 if (!hasNext()) throw
NoSuchElementException()\n
 last = index++\n
 return get(last)\n
 }\n
 override fun remove()
{\n
 check(last != -1) { "Call next() or previous() before removing element from the iterator." }\n
 removeAt(last)\n
 index = last\n
 last = -1\n
 }\n
 }\n
 /**\n
 * Implementation of
`MutableListIterator` for abstract lists.\n
 * \n
 private inner class ListIteratorImpl(index: Int) : IteratorImpl(),
MutableListIterator<E> {\n
 init {\n
 AbstractList.checkPositionIndex(index,
this@AbstractMutableList.size)\n
 this.index = index\n
 }\n
 override fun hasPrevious(): Boolean =
index > 0\n
 override fun nextIndex(): Int = index\n
 override fun previous(): E {\n
 if
(!hasPrevious()) throw NoSuchElementException()\n
 last = --index\n
 return get(last)\n
 }\n
 override fun previousIndex(): Int = index - 1\n
 override fun add(element: E) {\n
 add(index, element)\n
 index++\n
 last = -1\n
 }\n
 override fun set(element: E) {\n
 check(last != -1) { "Call
next() or previous() before updating element value with the iterator." }\n
 set(last, element)\n
 }\n
 }\n
 private class SubList<E>(private val list: AbstractMutableList<E>, private val fromIndex: Int, toIndex: Int) :
AbstractMutableList<E>(), RandomAccess {\n
 private var _size: Int = 0\n
 init {\n
 AbstractList.checkRangeIndexes(fromIndex, toIndex, list.size)\n
 this._size = toIndex - fromIndex\n
 }\n
 override fun add(index: Int, element: E) {\n
 AbstractList.checkPositionIndex(index, _size)\n
 list.add(fromIndex + index, element)\n
 _size++\n
 }\n
 override fun get(index: Int): E {\n
 AbstractList.checkElementIndex(index, _size)\n
 return list[fromIndex + index]\n
 }\n
 override
fun removeAt(index: Int): E {\n
 AbstractList.checkElementIndex(index, _size)\n
 val result =
list.removeAt(fromIndex + index)\n
 _size--\n
 return result\n
 }\n
 override fun set(index: Int,
element: E): E {\n
 AbstractList.checkElementIndex(index, _size)\n
 return list.set(fromIndex + index,
element)\n
 }\n
 override val size: Int get() = _size\n
 internal override fun checkIsMutable(): Unit =
list.checkIsMutable()\n
 }\n
}\n
}"/\n
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
 */\n
 * Based on GWT AbstractMap\n
 * Copyright 2007 Google Inc.\n
 */\n
\n
package kotlin.collections\n
/**\n
 * Provides a skeletal implementation of the [MutableMap] interface.\n
 * \n
 * The implementor is required to implement [entries] property, which should return mutable set of map entries, and
[put] function.\n
 * \n
 * @param K the type of map keys. The map is invariant in its key type.\n
 * \n
 * @param V the type
of map values. The map is invariant in its value type.\n
 */\n
\n
public actual abstract class AbstractMutableMap<K, V>
protected actual constructor() : AbstractMap<K, V>(), MutableMap<K, V> {\n
 /**\n
 * A mutable
[Map.Entry] shared by several [Map] implementations.\n
 * \n
 internal open class SimpleEntry<K, V>(override
val key: K, value: V) : MutableMap.MutableEntry<K, V> {\n
 constructor(entry: Map.Entry<K, V>) :
this(entry.key, entry.value)\n
 private var _value = value\n
 override val value: V get() = _value\n
 override fun setValue(newValue: V): V {\n
 // Should check if the map containing this entry is mutable.\n
 // However, to not increase entry memory footprint it might be worthwhile not to check it here and\n
 //
force subclasses that implement `build()` (freezing) operation to implement their own `MutableEntry`.\n
//
this@AbstractMutableMap.checkIsMutable()\n
 val oldValue = this._value\n
 this._value = newValue\n
 return oldValue\n
 }\n
 }\n
 override fun hashCode(): Int = entryHashCode(this)\n
 override fun
toString(): String = entryToString(this)\n
 override fun equals(other: Any?): Boolean = entryEquals(this,
other)\n
 }\n
}\n
 // intermediate abstract class to workaround KT-43321\n
 internal abstract class

```



```

AbstractEntrySet<E : Map.Entry<K, V>, K, V> : AbstractMutableSet<E>() {\n    final override fun
contains(element: E): Boolean = containsEntry(element)\n    abstract fun containsEntry(element: Map.Entry<K,
V>): Boolean\n    final override fun remove(element: E): Boolean = removeEntry(element)\n    abstract fun
removeEntry(element: Map.Entry<K, V>): Boolean\n    }\n\n    actual override fun clear() {\n    entries.clear()\n    }\n\n    private var _keys: MutableSet<K>? = null\n    actual override val keys: MutableSet<K>\n    get() {\n    if (_keys == null) {\n        _keys = object : AbstractMutableSet<K>() {\n            override fun
add(element: K): Boolean = throw UnsupportedOperationException("Add is not supported on keys")\n            override fun clear() {\n                this@AbstractMutableMap.clear()\n            }\n            override
operator fun contains(element: K): Boolean = containsKey(element)\n            override operator fun iterator():
MutableIterator<K> {\n                val entryIterator = entries.iterator()\n                return object :
MutableIterator<K> {\n                    override fun hasNext(): Boolean = entryIterator.hasNext()\n                    override fun next(): K = entryIterator.next().key\n                    override fun remove() =
entryIterator.remove()\n                }\n            }\n            override fun remove(element: K): Boolean
{\n                checkIsMutable()\n                if (containsKey(element)) {\n
this@AbstractMutableMap.remove(element)\n                return true\n            }\n            return
false\n        }\n        override val size: Int get() = this@AbstractMutableMap.size\n        override fun
checkIsMutable(): Unit = this@AbstractMutableMap.checkIsMutable()\n    }\n    }\n    return _keys!!\n    }\n\n    actual abstract override fun put(key: K, value: V): V?\n    actual override fun
putAll(from: Map<out K, V>) {\n    checkIsMutable()\n    for ((key, value) in from) {\n        put(key,
value)\n    }\n    }\n\n    private var _values: MutableCollection<V>? = null\n    actual override val values:
MutableCollection<V>\n    get() {\n    if (_values == null) {\n        _values = object :
AbstractMutableCollection<V>() {\n            override fun add(element: V): Boolean = throw
UnsupportedOperationException("Add is not supported on values")\n            override fun clear() =
this@AbstractMutableMap.clear()\n            override operator fun contains(element: V): Boolean =
containsValue(element)\n            override operator fun iterator(): MutableIterator<V> {\n                val
entryIterator = entries.iterator()\n                return object : MutableIterator<V> {\n                    override fun
hasNext(): Boolean = entryIterator.hasNext()\n                    override fun next(): V = entryIterator.next().value\n                    override fun remove() = entryIterator.remove()\n                }\n            }\n        }\n        override val size: Int get() = this@AbstractMutableMap.size\n        override fun
checkIsMutable(): Unit =
this@AbstractMutableMap.checkIsMutable()\n    }\n    }\n    return _values!!\n    }\n\n    actual
override fun remove(key: K): V? {\n    checkIsMutable()\n    val iter = entries.iterator()\n    while
(iter.hasNext()) {\n        val entry = iter.next()\n        val k = entry.key\n        if (key == k) {\n            val
value = entry.value\n            iter.remove()\n            return value\n        }\n    }\n    return null\n    }\n\n    /**\n     * This method is called every time when a mutating method is called on this mutable map.\n     * Mutable
maps that are built (frozen) must throw `UnsupportedOperationException`.\n     */\n    internal open fun
checkIsMutable(): Unit {\n    }\n\n    /**\n     * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n     * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n     */\n    package kotlin.collections\n\n    /**\n     * Provides a skeletal implementation of the
[MutableSet] interface.\n     */\n    @param E the type of elements contained in the set. The set is invariant in its
element type.\n\n    public actual abstract class AbstractMutableSet<E> protected actual constructor() :
AbstractMutableCollection<E>(), MutableSet<E> {\n\n        /**\n         * Compares this set with another set instance with
the unordered structural equality.\n         */\n        @return `true`, if [other] instance is a [Set] of the same size, all
elements of which are contained in this set.\n         */\n        override fun equals(other: Any?): Boolean {\n            if (other
=== this) return true\n            if (other !is Set<*>) return false\n            return AbstractSet.setEquals(this, other)\n        }\n\n        /**\n         * Returns the hash code value for this set.\n         */\n        override fun hashCode(): Int =
AbstractSet.unorderedHashCode(this)\n    }\n\n    /**\n     * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n     * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n     */\n    package kotlin.collections\n\n    /**\n     * Provides a [MutableList] implementation,

```

which uses a resizable array as its backing storage.

* This implementation doesn't provide a way to manage capacity, as backing JS array is resizeable itself.

* There is no speed advantage to pre-allocating array sizes in JavaScript, so this implementation does not include any of the "capacity and "growth increment" concepts.

```

public actual open class ArrayList<E> internal constructor(private var array: Array<Any?>) :
AbstractMutableList<E>(), MutableList<E>, RandomAccess {
    private var isReadOnly: Boolean = false
    /**
     * Creates an empty [ArrayList].
     */
    public actual constructor() : this(emptyArray()) {}
    /**
     * Creates an empty [ArrayList].
     * @param initialCapacity initial capacity (ignored)
     */
    public actual constructor(initialCapacity: Int) : this(emptyArray()) {}
    /**
     * Creates an [ArrayList] filled from the [elements] collection.
     */
    public actual constructor(elements: Collection<E>) :
this(elements.toArray<Any?>()) {}
    @PublishedApi
    internal fun build(): List<E> {
        checkIsMutable()
        isReadOnly = true
        return this
    }
    /** Does nothing in this ArrayList implementation.
     */
    public actual fun trimToSize() {}
    /** Does nothing in this ArrayList implementation.
     */
    public actual fun ensureCapacity(minCapacity: Int) {}
    actual override val size: Int get() = array.size
    @Suppress("UNCHECKED_CAST")
    actual override fun get(index: Int): E = array[rangeCheck(index)] as E
    actual override fun set(index: Int, element: E): E {
        checkIsMutable()
        rangeCheck(index)
        @Suppress("UNCHECKED_CAST")
        return array[index].apply { array[index] = element } as E
    }
    actual override fun add(element: E): Boolean {
        checkIsMutable()
        array.asDynamic().push(element)
        modCount++
        return true
    }
    actual override fun add(index: Int, element: E): Unit {
        checkIsMutable()
        array.asDynamic().splice(insertionRangeCheck(index), 0, element)
        modCount++
    }
    actual override fun addAll(elements: Collection<E>): Boolean {
        checkIsMutable()
        if (elements.isEmpty()) return false
        array += elements.toArray<Any?>()
        modCount++
        return true
    }
    actual override fun addAll(index: Int, elements: Collection<E>): Boolean {
        checkIsMutable()
        insertionRangeCheck(index)
        if (index == size) return addAll(elements)
        if (elements.isEmpty()) return false
        when (index) {
            size -> return addAll(elements)
            0 -> array = elements.toArray<Any?>() + array
            else -> array = array.copyOfRange(0, index).asDynamic().concat(elements.toArray<Any?>(), array.copyOfRange(index, size))
        }
        modCount++
        return true
    }
    actual override fun removeAt(index: Int): E {
        checkIsMutable()
        rangeCheck(index)
        modCount++
        return if (index == lastIndex) array.asDynamic().pop()
        else array.asDynamic().splice(index, 1)[0]
    }
    actual override fun remove(element: E): Boolean {
        checkIsMutable()
        for (index in array.indices) {
            if (array[index] == element) {
                array.asDynamic().splice(index, 1)
                modCount++
                return true
            }
        }
        return false
    }
    override fun removeRange(fromIndex: Int, toIndex: Int) {
        checkIsMutable()
        modCount++
        array.asDynamic().splice(fromIndex, toIndex - fromIndex)
    }
    actual override fun clear() {
        checkIsMutable()
        array = emptyArray()
        modCount++
    }
    actual override fun indexOf(element: E): Int = array.indexOf(element)
    actual override fun lastIndexOf(element: E): Int = array.lastIndexOf(element)
    override fun toString() = arrayToString(array)
    @Suppress("UNCHECKED_CAST")
    override fun <T> toArray(array: Array<T>): Array<T> {
        if (array.size < size) {
            return toArray() as Array<T>
        }
        (this.array as Array<T>).copyInto(array)
        if (array.size > size) {
            array[size] = null as T // null-terminate
        }
        return array
    }
    override fun toArray(): Array<Any?> {
        return js("[]").slice.call(array)
    }
    internal override fun checkIsMutable() {
        if (isReadOnly) throw UnsupportedOperationException()
    }
    private fun rangeCheck(index: Int) = index.apply {
        AbstractList.checkElementIndex(index, size)
    }
    private fun insertionRangeCheck(index: Int) = index.apply {
        AbstractList.checkPositionIndex(index, size)
    }
}

```

"/>**\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\ninternal fun <T> sortArrayWith(array: Array<out T>, comparison: (T, T) -> Int) {\n if (getStableSortingIsSupported()) {\n array.asDynamic().sort(comparison)\n } else {\n mergeSort(array.unsafeCast<Array<T>>(), 0, array.lastIndex, Comparator(comparison))\n }\n}\n\ninternal fun

```

<T> sortArrayWith(array: Array<out T>, comparator: Comparator<in T>) {\n  if (getStableSortingIsSupported())
{\n    val comparison = { a: T, b: T -> comparator.compare(a, b) }\n    array.asDynamic().sort(comparison)\n  } else {\n    mergeSort(array.unsafeCast<Array<T>>(), 0, array.lastIndex, comparator)\n  }\n}\n\ninternal fun
<T> sortArrayWith(array: Array<out T>, fromIndex: Int, toIndex: Int, comparator: Comparator<in T>) {\n  if
(fromIndex < toIndex - 1) {\n    mergeSort(array.unsafeCast<Array<T>>(), fromIndex, toIndex - 1, comparator)\n
  }\n}\n\ninternal fun <T : Comparable<T>> sortArray(array: Array<out T>) {\n  if
(getStableSortingIsSupported()) {\n    val comparison = { a: T, b: T -> a.compareTo(b) }\n
array.asDynamic().sort(comparison)\n  } else {\n    mergeSort(array.unsafeCast<Array<T>>(), 0,
array.lastIndex, naturalOrder())\n  }\n}\n\nprivate var _stableSortingIsSupported: Boolean? = null\nprivate fun
getStableSortingIsSupported(): Boolean {\n  _stableSortingIsSupported?.let { return it }\n
_stableSortingIsSupported = false\n  val array = js("[ ]").unsafeCast<Array<Int>>()\n  // known
implementations may use stable sort for arrays of up to 512 elements\n  // so we create slightly more elements to
test stability\n  for (index in 0 until 600) array.asDynamic().push(index)\n  val comparison = { a: Int, b: Int -> (a
and 3) - (b and 3) }\n  array.asDynamic().sort(comparison)\n  for (index in 1 until array.size) {\n    val a =
array[index - 1]\n    val b = array[index]\n    if ((a and 3) == (b and 3) && a >= b) return false\n  }\n
_stableSortingIsSupported = true\n  return true\n}\n\nprivate fun <T> mergeSort(array: Array<T>, start: Int,
endInclusive: Int, comparator: Comparator<in T>) {\n  val buffer =
arrayOfNulls<Any?>(array.size).unsafeCast<Array<T>>()\n  val result = mergeSort(array, buffer, start,
endInclusive, comparator)\n  if (result !== array) {\n    for (i in start..endInclusive) array[i] = result[i]\n
  }\n}\n\n// Both start and end are inclusive indices.\nprivate fun <T> mergeSort(array: Array<T>, buffer: Array<T>,
start: Int, end: Int, comparator: Comparator<in T>): Array<T> {\n  if (start == end) {\n    return array\n  }\n
val median = (start + end) / 2\n  val left = mergeSort(array, buffer, start, median, comparator)\n  val right =
mergeSort(array, buffer, median + 1, end, comparator)\n  val target = if (left === buffer) array else buffer\n  //
Merge.\n  var leftIndex = start\n  var rightIndex = median + 1\n  for (i in start..end) {\n    when {\n
leftIndex <= median && rightIndex <= end -> {\n      val leftValue = left[leftIndex]\n      val rightValue
= right[rightIndex]\n      if (comparator.compare(leftValue, rightValue) <= 0) {\n        target[i] =
leftValue\n        leftIndex++\n      } else {\n        target[i] = rightValue\n
rightIndex++\n      }\n    }\n    leftIndex <= median -> {\n      target[i] = left[leftIndex]\n
leftIndex++\n    }\n    else /* rightIndex <= end */ -> {\n      target[i] = right[rightIndex]\n
rightIndex++\n      Unit // TODO: Fix KT-31506\n    }\n  }\n  return target\n}"/*\n *
Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.collections\n\n@OptIn(ExperimentalUnsignedTypes::class)\n@SinceKotlin("1.3")\n@kotlin.js.JsName("
contentDeepHashCodeImpl")\ninternal fun <T> Array<out T>?.contentDeepHashCodeImpl(): Int {\n  if (this ==
null) return 0\n  var result = 1\n  for (element in this) {\n    val elementHash = when {\n      element == null
-> 0\n      isArrayish(element) -> (element.unsafeCast<Array<*>>()).contentDeepHashCodeImpl()\n
element is UByteArray -> element.contentHashCode()\n      element is UShortArray ->
element.contentHashCode()\n      element is UIntArray -> element.contentHashCode()\n      element is
ULongArray -> element.contentHashCode()\n      else -> element.hashCode()\n    }\n    result = 31 * result + elementHash\n  }\n  return result\n}"/*\n * Copyright 2010-2018 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\ninternal interface
EqualityComparator {\n  /**\n   * Subclasses must override to return a value indicating\n   * whether or not two
keys or values are equal.\n   */\n  abstract fun equals(value1: Any?, value2: Any?): Boolean\n\n  /**\n   *
Subclasses must override to return the hash code of a given key.\n   */\n  abstract fun getHashCode(value: Any?):
Int\n}\n\nobject HashCode : EqualityComparator {\n  override fun equals(value1: Any?, value2: Any?):
Boolean = value1 == value2\n  override fun getHashCode(value: Any?): Int = value?.hashCode() ?: 0\n
}\n}"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this

```

```

source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
Based on GWT AbstractHashMap
Copyright 2008 Google Inc.
package kotlin.collections
import
kotlin.collections.MutableMap.MutableEntry
Hash table based implementation of the [MutableMap]
interface.
This implementation makes no guarantees regarding the order of enumeration of [keys], [values]
and [entries] collections.
// Classes that extend HashMap and implement `build()` (freezing) operation
// have to make sure mutating methods check `checkIsMutable`.
public actual open class HashMap<K, V> :
AbstractMutableMap<K, V>, MutableMap<K, V> {
    private inner class EntrySet :
AbstractEntrySet<MutableEntry<K, V>, K, V>() {
        override fun add(element: MutableEntry<K, V>):
Boolean = throw UnsupportedOperationException("Add is not supported on entries")
        override fun clear() {
            this@HashMap.clear()
        }
        override fun containsEntry(element: Map.Entry<K, V>): Boolean =
this@HashMap.containsEntry(element)
        override operator fun iterator():
MutableIterator<MutableEntry<K, V>> = internalMap.iterator()
        override fun removeEntry(element:
Map.Entry<K, V>): Boolean {
            if (contains(element)) {
                this@HashMap.remove(element.key)
            }
            return true
        }
        override val size: Int get() =
this@HashMap.size
    }
    /**
     * Internal implementation of the map: either string-based or hashcode-
based.
     */
    private val internalMap: InternalMap<K, V>
    private val equality: EqualityComparator
    internal constructor(internalMap: InternalMap<K, V>) : super() {
        this.internalMap = internalMap
        this.equality = internalMap.equality
    }
    /**
     * Constructs an empty [HashMap] instance.
     */
    actual constructor() : this(InternalHashMap(EqualityComparator.HashCode))
    /**
     * Constructs an
empty [HashMap] instance.
     */
    @param initialCapacity the initial capacity (ignored)
    @param
loadFactor the load factor (ignored)
     */
    @throws IllegalArgumentException if the initial capacity or
load factor are negative
    actual constructor(initialCapacity: Int, loadFactor: Float) : this() {
        // This
implementation of HashMap has no need of load factors or capacities.
        require(initialCapacity >= 0) {
            "Negative initial capacity: $initialCapacity"
        }
        require(loadFactor >= 0) {
            "Non-positive load factor:
$loadFactor"
        }
    }
    actual constructor(initialCapacity: Int) : this(initialCapacity, 0.0f)
    /**
     *
Constructs an instance of [HashMap] filled with the contents of the specified [original] map.
     */
    actual
constructor(original: Map<out K, V>) : this() {
        this.putAll(original)
    }
    actual override fun clear() {
        internalMap.clear()
    }
    // structureChanged(this)
    actual override fun containsKey(key: K): Boolean =
internalMap.containsKey()
    actual override fun containsValue(value: V): Boolean = internalMap.any {
        equality.equals(it.value, value)
    }
    private var _entries: MutableSet<MutableMap.MutableEntry<K, V>>? =
null
    actual override val entries: MutableSet<MutableMap.MutableEntry<K, V>>
get() {
        if
(_entries == null) {
            _entries = createEntrySet()
        }
        return _entries!!
    }
    internal
open fun createEntrySet(): MutableSet<MutableMap.MutableEntry<K, V>> = EntrySet()
    actual override
operator fun get(key: K): V? = internalMap.get(key)
    actual override fun put(key: K, value: V): V? =
internalMap.put(key, value)
    actual override fun remove(key: K): V? = internalMap.remove(key)
    actual
override val size: Int get() = internalMap.size
}
/**
 * Constructs the specialized implementation of
[HashMap] with [String] keys, which stores the keys as properties of
a JS object without hashing them.
 */
public fun <V> stringMapOf(vararg pairs: Pair<String, V>): HashMap<String, V> {
    return
HashMap<String, V>(InternalStringMap(EqualityComparator.HashCode)).apply { putAll(pairs) }
}
Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
Based on GWT
HashSet
Copyright 2008 Google Inc.
package kotlin.collections
The implementation of the
[MutableSet] interface, backed by a [HashMap] instance.
// Classes that extend HashSet and implement
`build()` (freezing) operation
// have to make sure mutating methods check `checkIsMutable`.
public actual open
class HashSet<E> : AbstractMutableSet<E>, MutableSet<E> {
    internal val map: HashMap<E, Any>
    /**
     * Constructs a new empty [HashSet].
     */
    actual constructor() {
        map = HashMap<E, Any>()
    }
    /**
     * Constructs a new [HashSet] filled with the elements of the specified collection.
     */
    actual
constructor(elements: Collection<E>) {
        map = HashMap<E, Any>(elements.size)
        addAll(elements)
    }
}

```



```

getChainOrEntryOrNull(equality.getHashCode(key)) ?: return null\n    if (chainOrEntry !is Array<*>) {\n
val entry: MutableEntry<K, V> = chainOrEntry\n    if (equality.equals(entry.key, key)) {\n    return
entry\n    } else {\n    return null\n    }\n    } else {\n    val chain: Array<MutableEntry<K,
V>> = chainOrEntry\n    return chain.findEntryInChain(key)\n    }\n    }\n\n private fun
Array<MutableEntry<K, V>>.findEntryInChain(key: K): MutableEntry<K, V>? =\n    firstOrNull { entry ->
equality.equals(entry.key, key) }\n\n override fun iterator(): MutableIterator<MutableEntry<K, V>> {\n\n
return object : MutableIterator<MutableEntry<K, V>> {\n    var state = -1 // -1 not ready, 0 - ready, 1 -
done\n\n    val keys: Array<String> = js("Object").keys(backingMap)\n    var keyIndex = -1\n\n
var chainOrEntry: dynamic = null\n    var isChain = false\n    var itemIndex = -1\n    var lastEntry:
MutableEntry<K, V>? = null\n\n    private fun computeNext(): Int {\n    if (chainOrEntry != null &&
isChain) {\n    val chainSize: Int = chainOrEntry.unsafeCast<Array<MutableEntry<K, V>>>().size\n
    if (++itemIndex < chainSize)\n    return 0\n    }\n    if (++keyIndex < keys.size)\n
{\n    chainOrEntry = backingMap[keys[keyIndex]]\n    isChain = chainOrEntry is Array<*>\n
    itemIndex = 0\n    return 0\n    } else {\n    chainOrEntry = null\n
return 1\n    }\n    }\n\n    override fun hasNext(): Boolean {\n    if (state == -1)\n
state = computeNext()\n    return state == 0\n    }\n\n    override fun next(): MutableEntry<K, V>
{\n    if (!hasNext()) throw NoSuchElementException()\n    val lastEntry = if (isChain) {\n
chainOrEntry.unsafeCast<Array<MutableEntry<K, V>>>()[itemIndex]\n    } else {\n
chainOrEntry.unsafeCast<MutableEntry<K, V>>()\n    }\n    this.lastEntry = lastEntry\n
state = -1\n    return lastEntry\n    }\n\n    override fun remove() {\n
checkNotNull(lastEntry)\n    this@InternalHashMap.remove(lastEntry!!.key)\n    lastEntry =
null\n    // the chain being iterated just got modified by InternalHashMap.remove\n    itemIndex-
-1\n    }\n    }\n\n    private fun getChainOrEntryOrNull(hashCode: Int): dynamic {\n    val
chainOrEntry = backingMap[hashCode]\n    return if (chainOrEntry === undefined) null else chainOrEntry\n
}\n\n}\n\n", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n */\npackage kotlin.collections\n\n/**\n * The common interface of [InternalStringMap] and
[InternalHashMap].\n */\ninternal interface InternalMap<K, V> :
MutableIterable<MutableMap.MutableEntry<K, V>> {\n    val equality: EqualityComparator\n    val size: Int\n
operator fun contains(key: K): Boolean\n    operator fun get(key: K): V?\n\n    fun put(key: K, value: V): V?\n
    fun remove(key: K): V?\n    fun clear(): Unit\n\n    fun createJsMap(): dynamic {\n    val result =
js("Object.create(null)")\n    // force to switch object representation to dictionary mode\n    result["foo"] =
1\n    jsDeleteProperty(result, "foo")\n    return result\n    }\n\n}\n\n", /*\n * Copyright 2010-2018 JetBrains s.r.o.
and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n */\n\n * Based on GWT InternalStringMap\n * Copyright
2008 Google Inc.\n */\npackage kotlin.collections\n\nimport kotlin.collections.MutableMap.MutableEntry\n\n/**\n
* A simple wrapper around JavaScript Map for key type is string.\n */\n * Though this map is instantiated only with
K=String, the K type is not fixed to String statically,\n * because we want to have it erased to Any? in order not to
generate type-safe override bridges for\n * [get], [contains], [remove] etc, if they ever are generated.\n */\ninternal
class InternalStringMap<K, V>(override val equality: EqualityComparator) : InternalMap<K, V> {\n\n    private var
backingMap: dynamic = createJsMap()\n    override var size: Int = 0\n    private set\n\n    /**\n    * A mod
count to track 'value' replacements in map to ensure that the 'value' that we have in the\n    * iterator entry is
guaranteed to be still correct.\n    * This is to optimize for the common scenario where the values are not modified
during\n    * iterations where the entries are never stale.\n    */\n    private var valueMod: Int = 0\n\n    override
operator fun contains(key: K): Boolean {\n    if (key !is String) return false\n    return backingMap[key] !==
undefined\n    }\n\n    override operator fun get(key: K): V? {\n    if (key !is String) return null\n    val value =
backingMap[key]\n    return if (value !== undefined) value.unsafeCast<V>() else null\n    }\n\n    override fun
put(key: K, value: V): V? {\n    require(key is String)\n    val oldValue = backingMap[key]\n

```

```

backingMap[key] = value\n\n    if (oldValue === undefined) {\n        size++\n//
structureChanged(host)\n    return null\n    } else {\n//        valueMod++\n        return
oldValue.unsafeCast<V>()\n    }\n }\n\n override fun remove(key: K): V? {\n    if (key !is String) return
null\n    val value = backingMap[key]\n    if (value !== undefined) {\n        jsDeleteProperty(backingMap,
key)\n        size--\n//        structureChanged(host)\n        return value.unsafeCast<V>()\n    } else {\n//
valueMod++\n        return null\n    }\n }\n\n\n override fun clear() {\n    backingMap = createJsMap()\n
size = 0\n }\n\n\n override fun iterator(): MutableIterator<MutableEntry<K, V>> {\n    return object :
MutableIterator<MutableEntry<K, V>> {\n        private val keys: Array<String> =
js("Object").keys(backingMap)\n        private val iterator = keys.iterator()\n        private var lastKey: String? =
null\n\n        override fun hasNext(): Boolean = iterator.hasNext()\n\n        override fun next():
MutableEntry<K, V> {\n            val key = iterator.next()\n            lastKey = key\n\n            @Suppress("UNCHECKED_CAST")\n            return newMapEntry(key as K)\n        }\n\n        override
fun remove() {\n            @Suppress("UNCHECKED_CAST")\n            this@InternalStringMap.remove(checkNotNull(lastKey) as K)\n        }\n    }\n\n\n private fun
newMapEntry(key: K): MutableEntry<K, V> = object : MutableEntry<K, V> {\n        override val key: K get() =
key\n        override val value: V get() = this@InternalStringMap[key].unsafeCast<V>()\n\n        override fun
setValue(newValue: V): V = this@InternalStringMap.put(key, newValue).unsafeCast<V>()\n\n        override fun
hashCode(): Int = AbstractMap.entryHashCode(this)\n        override fun toString(): String =
AbstractMap.entryToString(this)\n        override fun equals(other: Any?): Boolean = AbstractMap.entryEquals(this,
other)\n    }\n }\n\n\n", /*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n *
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n * Based on GWT LinkedHashMap\n * Copyright 2008 Google Inc.\n */\n * package
kotlin.collections\n\nimport kotlin.collections.MutableMap.MutableEntry\n\n/**\n * Hash table based
implementation of the [MutableMap] interface, which additionally preserves the insertion order\n * of entries during
the iteration.\n * The insertion order is preserved by maintaining a doubly-linked list of all of its entries.\n
*/\n * public actual open class LinkedHashMap<K, V> : HashMap<K, V>, MutableMap<K, V> {\n\n    /**\n     * The
entry we use includes next/prev pointers for a doubly-linked circular\n     * list with a head node. This reduces the
special cases we have to deal with\n     * in the list operations.\n     * Note that we duplicate the key from the
underlying hash map so we can find\n     * the eldest entry. The alternative would have been to modify HashMap so
more\n     * of the code was directly usable here, but this would have added some\n     * overhead to HashMap, or to
reimplement most of the HashMap code here with\n     * small modifications. Paying a small storage cost only if
you use\n     * LinkedHashMap and minimizing code size seemed like a better tradeoff\n     */\n     private inner class
ChainEntry<K, V>(key: K, value: V) : AbstractMutableMap.SimpleEntry<K, V>(key, value) {\n        internal var
next: ChainEntry<K, V>? = null\n        internal var prev: ChainEntry<K, V>? = null\n\n        override fun
setValue(newValue: V): V {\n            this@LinkedHashMap.checkIsMutable()\n            return
super.setValue(newValue)\n        }\n    }\n\n    private inner class EntrySet : AbstractEntrySet<MutableEntry<K,
V>, K, V>() {\n        private inner class EntryIterator : MutableIterator<MutableEntry<K, V>> {\n            // The
last entry that was returned from this iterator.\n            private var last: ChainEntry<K, V>? = null\n            // The
next entry to return from this iterator.\n            private var next: ChainEntry<K, V>? = null\n\n            init {\n
                next = head\n//                recordLastKnownStructure(map, this)\n            }\n\n            override fun hasNext():
Boolean {\n                return next !== null\n            }\n\n            override fun next(): MutableEntry<K, V> {\n//
                checkStructuralChange(map, this)\n                if (!hasNext()) throw NoSuchElementException()\n\n                val
current = next!!\n                last = current\n                next = current.next.takeIf { it !== head }\n                return
current\n            }\n\n            override fun remove() {\n                check(last !== null)\n                this@EntrySet.checkIsMutable()\n//                checkStructuralChange(map, this)\n                last!!.remove()\n
                map.remove(last!!.key)\n//                recordLastKnownStructure(map, this)\n                last = null\n            }\n
        }\n\n        override fun add(element: MutableEntry<K, V>): Boolean = throw
UnsupportedOperationException("Add is not supported on entries")\n\n        override fun clear() {\n

```

```

this@LinkedHashMap.clear()\n    }\n    override fun containsEntry(element: Map.Entry<K, V>): Boolean =
this@LinkedHashMap.containsEntry(element)\n    override operator fun iterator():
MutableIterator<MutableEntry<K, V>> = EntryIterator()\n    override fun removeEntry(element: Map.Entry<K,
V>): Boolean {\n        checkIsMutable()\n        if (contains(element)) {\n
this@LinkedHashMap.remove(element.key)\n            return true\n        }\n        return false\n    }\n
override val size: Int get() = this@LinkedHashMap.size\n    override fun checkIsMutable(): Unit =
this@LinkedHashMap.checkIsMutable()\n    }\n\n    /*\n     * The head of the insert order chain, which is a doubly-
linked circular\n     * list.\n     */\n    * The most recently inserted node is at the end of the chain, ie.\n     * chain.prev.\n     */\n    private var head: ChainEntry<K, V>? = null\n\n    /**\n     * Add this node to the end of the chain.\n     */\n    private fun ChainEntry<K, V>.addToEnd() {\n        // This entry is not in the list.\n        check(next == null && prev
== null)\n        val _head = head\n        if (_head == null) {\n            head = this\n            next = this\n            prev =
this\n        } else {\n            // Chain is valid.\n            val _tail = checkNotNull(_head.prev)\n            // Update me.\n            prev = _tail\n            next = _head\n            // Update my new siblings: current head and old tail\n            _head.prev = this\n            _tail.next = this\n        }\n    }\n\n    /**\n     * Remove this node from the chain it is a part
of.\n     */\n    private fun ChainEntry<K, V>.remove() {\n        if (this.next === this) {\n            // if this is single
element, remove head\n            head = null\n        } else {\n            if (head === this) {\n                // if this is first
element, move head to next\n                head = next\n            }\n            next!!.prev = prev\n            prev!!.next =
next\n        }\n        next = null\n        prev = null\n    }\n\n    /*\n     * The hashmap that keeps track of our entries and
the chain. Note that we\n     * duplicate the key here to eliminate changes to HashMap and minimize the\n     * code
here, at the expense of additional space.\n     */\n    private val map: HashMap<K, ChainEntry<K, V>>\n\n    private
var isReadOnly: Boolean = false\n\n    /**\n     * Constructs an empty [LinkedHashMap] instance.\n     */\n    actual
constructor(): super() {\n        map = HashMap<K, ChainEntry<K, V>>()\n    }\n\n    internal
constructor(backingMap: HashMap<K, Any>): super() {\n        @Suppress("UNCHECKED_CAST") // expected
to work due to erasure\n        map = backingMap as HashMap<K, ChainEntry<K, V>>\n    }\n\n    /**\n     *
Constructs an empty [LinkedHashMap] instance.\n     */\n    * @param initialCapacity the initial capacity
(ignored)\n     * @param loadFactor the load factor (ignored)\n     */\n    * @throws IllegalArgumentException if
the initial capacity or load factor are negative\n     */\n    actual constructor(initialCapacity: Int, loadFactor: Float) :
super(initialCapacity, loadFactor) {\n        map = HashMap<K, ChainEntry<K, V>>()\n    }\n\n    actual
constructor(initialCapacity: Int) : this(initialCapacity, 0.0f)\n\n    /**\n     * Constructs an instance of
[LinkedHashMap] filled with the contents of the specified [original] map.\n     */\n    actual constructor(original:
Map<out K, V>) {\n        map = HashMap<K, ChainEntry<K, V>>()\n        this.putAll(original)\n    }\n\n    @PublishedApi
internal fun build(): Map<K, V> {\n        checkIsMutable()\n        isReadOnly = true\n        return this\n    }\n\n    actual override fun clear() {\n        checkIsMutable()\n        map.clear()\n        head = null\n    }\n\n    // override fun clone(): Any {\n        // return LinkedHashMap(this)\n    }\n\n    actual override fun
containsKey(key: K): Boolean = map.containsKey(key)\n\n    actual override fun containsValue(value: V): Boolean
{\n        var node: ChainEntry<K, V> = head ?: return false\n        do {\n            if (node.value == value) {\n                return true\n            }\n            node = node.next!!\n        } while (node !== head)\n        return false\n    }\n\n    internal override fun createEntrySet(): MutableSet<MutableMap.MutableEntry<K, V>> = EntrySet()\n\n    actual
override operator fun get(key: K): V? = map.get(key)?.value\n\n    actual override fun put(key: K, value: V): V? {\n        checkIsMutable()\n        val old = map.get(key)\n        if (old == null) {\n            val newEntry =
ChainEntry(key, value)\n            map.put(key, newEntry)\n            newEntry.addToEnd()\n            return null\n        } else {\n            return old.setValue(value)\n        }\n    }\n\n    actual override fun remove(key: K): V? {\n        checkIsMutable()\n        val entry = map.remove(key)\n        if (entry != null) {\n            entry.remove()\n            return entry.value\n        }\n        return null\n    }\n\n    actual override val size: Int get() = map.size\n\n    internal
override fun checkIsMutable() {\n        if (isReadOnly) throw UnsupportedOperationException()\n    }\n}\n\n/**\n     *
Constructs the specialized implementation of [LinkedHashMap] with [String] keys, which stores the keys as
properties of\n     * JS object without hashing them.\n     */\n    public fun <V> linkedStringMapOf(vararg pairs:
Pair<String, V>): LinkedHashMap<String, V> {\n        return LinkedHashMap<String,

```



```

*\n@jsName("output")\ninternal var output = run {\n    val isNode: Boolean = js("typeof process !== 'undefined'
&& process.versions && !process.versions.node")\n    if (isNode) NodeJsOutput(js("process.stdout")) else
BufferedOutputToConsoleLog()\n}\n\n@kotlin.internal.InlineOnly\nprivate inline fun String(value: Any?): String =
js("String")(value)\n\n/** Prints the line separator to the standard output stream. */\npublic actual fun println() {\n
output.println()\n}\n\n/** Prints the given [message] and the line separator to the standard output stream. */\npublic
actual fun println(message: Any?) {\n    output.println(message)\n}\n\n/** Prints the given [message] to the standard
output stream. */\npublic actual fun print(message: Any?) {\n
output.print(message)\n}\n\n@SinceKotlin("1.6")\npublic actual fun readln(): String = throw
UnsupportedOperationException("readln is not supported in Kotlin/JS")\n\n@SinceKotlin("1.6")\npublic actual
fun readlnOrNull(): String? = throw UnsupportedOperationException("readlnOrNull is not supported in
Kotlin/JS"), /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use
of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\npackage kotlin.coroutines\n\nimport kotlin.coroutines.intrinsics.CoroutineSingletons.*\nimport
kotlin.coroutines.intrinsics.COROUTINE_SUSPENDED\n\n@PublishedApi\n@SinceKotlin("1.3")\ninternal
actual class SafeContinuation<in T>\ninternal actual constructor(\n    private val delegate: Continuation<T>,\n
initialResult: Any?\n) : Continuation<T> {\n    @PublishedApi\n    internal actual constructor(delegate:
Continuation<T>) : this(delegate, UNDECIDED)\n\n    public actual override val context: CoroutineContext\n
get() = delegate.context\n\n    private var result: Any? = initialResult\n\n    public actual override fun
resumeWith(result: Result<T>) {\n        val cur = this.result\n        when {\n            cur === UNDECIDED -> {\n
                this.result = result.value\n            }\n            cur === COROUTINE_SUSPENDED -> {\n                this.result =
RESUMED\n                delegate.resumeWith(result)\n            }\n            else -> throw
IllegalStateException("Already resumed")\n        }\n    }\n\n    @PublishedApi\n    internal actual fun
getOrThrow(): Any? {\n        if (result === UNDECIDED) {\n            result = COROUTINE_SUSPENDED\n
return COROUTINE_SUSPENDED\n        }\n        val result = this.result\n        return when {\n            result ===
RESUMED -> COROUTINE_SUSPENDED // already called continuation, indicate COROUTINE_SUSPENDED
upstream\n            result is Result.Failure -> throw result.exception\n            else -> result // either
COROUTINE_SUSPENDED or data\n        }\n    }\n}\n\n", /*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n *\n\npackage
kotlin.coroutines.cancellation\n\n@SinceKotlin("1.4")\npublic actual open class CancellationException :
IllegalStateException {\n    actual constructor() : super()\n    actual constructor(message: String?) : super(message)\n
    constructor(message: String?, cause: Throwable?) : super(message, cause)\n    constructor(cause: Throwable?) :
super(cause)\n}\n\n", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n *
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\npackage kotlin.coroutines.js.internal\n\nimport kotlin.coroutines.Continuation\nimport
kotlin.coroutines.EmptyCoroutineContext\n\n@PublishedApi\n@SinceKotlin("1.3")\ninternal val
EmptyContinuation = Continuation<Any?>(EmptyCoroutineContext) { result -> result.getOrThrow()\n}\n\n", /*\n *
Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n\npackage
kotlin.js\n\n/**\n * Exposes the [Date API](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date) to Kotlin.\n
*\n\n@Suppress("NOT_DOCUMENTED")\npublic external class Date() {\n    public constructor(milliseconds:
Number)\n\n    public constructor(dateString: String)\n\n    public constructor(year: Int, month: Int)\n\n    public
constructor(year: Int, month: Int, day: Int)\n\n    public constructor(year: Int, month: Int, day: Int, hour: Int)\n\n
public constructor(year: Int, month: Int, day: Int, hour: Int, minute: Int)\n\n    public constructor(year: Int, month:
Int, day: Int, hour: Int, minute: Int, second: Int)\n\n    public constructor(year: Int, month: Int, day: Int, hour:
Int, minute: Int, second: Int, millisecond: Number)\n\n    public fun getDate(): Int\n\n    public fun getDay(): Int\n\n
public fun getFullYear(): Int\n\n    public fun getHours(): Int\n\n    public fun getMilliseconds(): Int\n\n    public fun

```

```

getMinutes(): Int\n\n public fun getMonth(): Int\n\n public fun getSeconds(): Int\n\n public fun getTime():
Double\n\n public fun getTimezoneOffset(): Int\n\n public fun getUTCDate(): Int\n\n public fun
getUTCDay(): Int\n\n public fun getUTCFullYear(): Int\n\n public fun getUTCHours(): Int\n\n public fun
getUTCMilliseconds(): Int\n\n public fun getUTCMinutes(): Int\n\n public fun getUTCMonth(): Int\n\n public
fun getUTCSeconds(): Int\n\n public fun toDateString(): String\n\n public fun toISOString(): String\n\n public
fun toJSON(): Json\n\n public fun toLocaleDateString(locales: Array<String> = definedExternally, options:
LocaleOptions = definedExternally): String\n\n public fun toLocaleDateString(locales: String, options:
LocaleOptions = definedExternally): String\n\n public fun toLocaleString(locales: Array<String> =
definedExternally, options: LocaleOptions = definedExternally): String\n\n public fun toLocaleString(locales:
String, options: LocaleOptions = definedExternally): String\n\n public fun toLocaleTimeString(locales:
Array<String> = definedExternally, options: LocaleOptions = definedExternally): String\n\n public fun
toLocaleTimeString(locales: String, options: LocaleOptions = definedExternally): String\n\n public fun
toTimeString(): String\n\n public fun toUTCString(): String\n\n public companion object {\n public fun
now(): Double\n\n public fun parse(dateString: String): Double\n\n public fun UTC(year: Int, month: Int):
Double\n\n public fun UTC(year: Int, month: Int, day: Int): Double\n\n public fun UTC(year: Int, month:
Int, day: Int, hour: Int): Double\n\n public fun UTC(year: Int, month: Int, day: Int, hour: Int, minute: Int):
Double\n\n public fun UTC(year: Int, month: Int, day: Int, hour: Int, minute: Int, second: Int): Double\n\n
public fun UTC(year: Int, month: Int, day: Int, hour: Int, minute: Int, second: Int, millisecond: Number): Double\n
}\n\n public interface LocaleOptions {\n public var localeMatcher: String?\n\n public var timeZone:
String?\n\n public var hour12: Boolean?\n\n public var formatMatcher: String?\n\n public var weekday:
String?\n\n public var era: String?\n\n public var year: String?\n\n public var month: String?\n\n
public var day: String?\n\n public var hour: String?\n\n public var minute: String?\n\n public var
second: String?\n\n public var timeZoneName: String?\n }\n}\n\npublic inline fun dateLocaleOptions(init:
Date.LocaleOptions.() -> Unit): Date.LocaleOptions {\n val result = js(\n"new
Object()\n").unsafeCast<Date.LocaleOptions>()\n init(result)\n return result\n}"/\n\n * Copyright 2010-2020
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.dom\n\nimport
org.w3c.dom.Document\n\nimport org.w3c.dom.Element\n\nimport
kotlin.internal.LowPriorityInOverloadResolution\n\nimport kotlinx.dom.appendChild as
newAppendElement\n\nimport kotlinx.dom.createElement as newCreateElement\n\n/**\n * Creates a new element
with the specified [name].\n * The element is initialized with the specified [init] function.\n
*/\n\n@LowPriorityInOverloadResolution\n@Deprecated(\n message = \n"This API is moved to another package,
use 'kotlinx.dom.createElement' instead.\n",\n replaceWith = ReplaceWith(\n"this.createElement(name, init)",
\n"\"kotlinx.dom.createElement()\"")\n)\n\n@DeprecatedSinceKotlin(warningSince = \n"1.4", errorSince = \n"1.6")\n\npublic
inline fun Document.createElement(name: String, noinline init: Element.() -> Unit): Element =
this.newCreateElement(name, init)\n\n/**\n * Appends a newly created element with the specified [name] to this
element.\n * The element is initialized with the specified [init] function.\n
*/\n\n@LowPriorityInOverloadResolution\n@Deprecated(\n message = \n"This API is moved to another package,
use 'kotlinx.dom.appendChild' instead.\n",\n replaceWith = ReplaceWith(\n"this.appendChild(name, init)",
\n"\"kotlinx.dom.appendChild()\"")\n)\n\n@DeprecatedSinceKotlin(warningSince = \n"1.4", errorSince = \n"1.6")\n\npublic
inline fun Element.appendChild(name: String, noinline init: Element.() -> Unit): Element =
this.newAppendElement(name, init)\n\n"/\n\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.dom\n\nimport org.w3c.dom.Element\n\nimport
kotlin.internal.LowPriorityInOverloadResolution\n\nimport kotlinx.dom.addClass as newAddClass\n\nimport
kotlinx.dom.hasClass as newHasClass\n\nimport kotlinx.dom.removeClass as newRemoveClass\n\n/** Returns true if
the element has the given CSS class style in its 'class' attribute
*/\n\n@LowPriorityInOverloadResolution\n@Deprecated(\n message = \n"This API is moved to another package,

```

```

use 'kotlinx.dom.hasClass' instead.\",\n    replaceWith = ReplaceWith(\"this.hasClass(cssClass)\",
\"kotlinx.dom.hasClass\")\n)\n\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\ninline fun
Element.hasClass(cssClass: String): Boolean = this.newHasClass(cssClass)\n\n/**\n * Adds CSS class to element.
Has no effect if all specified classes are already in class attribute of the element\n *\n * @return true if at least one
class has been added\n *\n@LowPriorityInOverloadResolution\n@Deprecated(\n    message = \"This API is moved
to another package, use 'kotlinx.dom.addClass' instead.\",\n    replaceWith =
ReplaceWith(\"this.addClass(cssClasses)\", \"kotlinx.dom.addClass\")\n)\n\n@DeprecatedSinceKotlin(warningSince
= \"1.4\", errorSince = \"1.6\")\ninline fun Element.addClass(vararg cssClasses: String): Boolean =
this.newAddClass(*cssClasses)\n\n/**\n * Removes all [cssClasses] from element. Has no effect if all specified
classes are missing in class attribute of the element\n *\n * @return true if at least one class has been removed\n
*\n@LowPriorityInOverloadResolution\n@Deprecated(\n    message = \"This API is moved to another package,
use 'kotlinx.dom.removeClass' instead.\",\n    replaceWith = ReplaceWith(\"this.removeClass(cssClasses)\",
\"kotlinx.dom.removeClass\")\n)\n\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\ninline
fun Element.removeClass(vararg cssClasses: String): Boolean = this.newRemoveClass(*cssClasses),\"/>\n\n *
Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n@npackage
kotlin.dom\n\nimport org.w3c.dom.Element\nimport org.w3c.dom.Node\nimport
kotlin.internal.LowPriorityInOverloadResolution\nimport kotlinx.dom.isElement as newIsElement\nimport
kotlinx.dom.isText as newIsText\n\n/**\n * Gets a value indicating whether this node is a TEXT_NODE or a
CDATA_SECTION_NODE.\n *\n@LowPriorityInOverloadResolution\n@Deprecated(\n    message = \"This API
is moved to another package, use 'kotlinx.dom.isText' instead.\",\n    replaceWith = ReplaceWith(\"this.isText\",
\"kotlinx.dom.isText\")\n)\n\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\npublic val
Node.isText: Boolean\n    inline get() = this.newIsText\n\n/**\n * Gets a value indicating whether this node is an
[Element].\n *\n@LowPriorityInOverloadResolution\n@Deprecated(\n    message = \"This API is moved to
another package, use 'kotlinx.dom.isElement' instead.\",\n    replaceWith = ReplaceWith(\"this.isElement\",
\"kotlinx.dom.isElement\")\n)\n\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\npublic val
Node.isElement: Boolean\n    inline get() = this.newIsElement\n\n\"/>\n\n * Copyright 2010-2018 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n *\n@npackage org.w3c.dom.events\n\npublic fun
EventListener(handler: (Event) -> Unit): EventListener = EventListenerHandler(handler)\n\nprivate class
EventListenerHandler(private val handler: (Event) -> Unit) : EventListener {\n    public override fun
handleEvent(event: Event) {\n        handler(event)\n    }\n\n    public override fun toString(): String =
\"EventListenerHandler($handler)\"\n}\n\n\"/>\n\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n *\n@npackage org.w3c.dom\n\npublic external interface ItemArrayLike<out T> {\n
val length: Int\n    fun item(index: Int): T?\n}\n\n/**\n * Returns the view of this `ItemArrayLike<T>` collection as
`List<T>`\n *\n@public fun <T> ItemArrayLike<T>.asList(): List<T> = object : AbstractList<T>() {\n    override val
size: Int get() = this@asList.length\n\n    override fun get(index: Int): T = when (index) {\n        in 0..lastIndex ->
this@asList.item(index).unsafeCast<T>()\n        else -> throw IndexOutOfBoundsException(\"index $index is not in
range [0..$lastIndex]\")\n    }\n}\n\n\"/>\n\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n *\n@npackage kotlin.dom\n\nimport org.w3c.dom.Element\nimport
org.w3c.dom.Node\nimport kotlin.internal.LowPriorityInOverloadResolution\nimport kotlinx.dom.appendText as
newAppendText\nimport kotlinx.dom.clear as newClear\n\n/**\n * Removes all the children from this node.\n
*\n@LowPriorityInOverloadResolution\n@Deprecated(\n    message = \"This API is moved to another package,
use 'kotlinx.dom.clear' instead.\",\n    replaceWith = ReplaceWith(\"this.clear()\",
\"kotlinx.dom.clear\")\n)\n\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\npublic inline fun
Node.clear() = this.newClear()\n\n/**\n * Creates text node and append it to the element.\n *\n * @return this

```

```

element\n *\n@LowPriorityInOverloadResolution\n@Deprecated(\n  message = \"This API is moved to another
package, use 'kotlinx.dom.appendText' instead.\",\n  replaceWith = ReplaceWith(\"this.appendText(text)\",
\"kotlinx.dom.appendText\")\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\ninline fun
Element.appendText(text: String): Element = this.newAppendText(text)\n\",/*\n * Copyright 2010-2018 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n *\n\npackage kotlin.js\n\n/**\n * Reinterprets this value
as a value of the [dynamic type](/docs/reference/dynamic-type.html).\n *\n\n@kotlin.internal.InlineOnly\npublic
inline fun Any?.asDynamic(): dynamic = this\n\n/**\n * Reinterprets this value as a value of the specified type [T]
without any actual type checking.\n *\n\n@kotlin.internal.InlineOnly\npublic inline fun <T> Any?.unsafeCast():
@kotlin.internal.NoInfer T = this.asDynamic()\n\n/**\n * Reinterprets this `dynamic` value as a value of the
specified type [T] without any actual type checking.\n
*\n\n@kotlin.internal.DynamicExtension\n@JsName(\"unsafeCastDynamic\")\n@kotlin.internal.InlineOnly\npublic
inline fun <T> dynamic.unsafeCast(): @kotlin.internal.NoInfer T = this\n\n/**\n * Allows to iterate this `dynamic`
object in the following cases:\n * - when it has an `iterator` function,\n * - when it is an array\n * - when it is an
instance of [kotlin.collections.Iterable]\n *\n\n@kotlin.internal.DynamicExtension\npublic operator fun
dynamic.iterator(): Iterator<dynamic> {\n  val r: Any? = this\n\n  return when {\n    this[\"iterator\"] != null -
->\n      this[\"iterator\"]()\n    isArrayish(r) ->\n      r.unsafeCast<Array<*>>().iterator()\n    else ->\n
      (r as Iterable<*>).iterator()\n  }\n}\n\",/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n *\n\n// a package is omitted to get declarations directly under the
module\n\n@JsName(\"throwNPE\")\ninternal fun throwNPE(message: String) {\n  throw
NullPointerException(message)\n}\n\n@JsName(\"throwCCE\")\ninternal fun throwCCE() {\n  throw
ClassCastException(\"Illegal cast\")\n}\n\n@JsName(\"throwISE\")\ninternal fun throwISE(message: String) {\n
throw IllegalStateException(message)\n}\n\n@JsName(\"throwUPAE\")\ninternal fun throwUPAE(propertyName:
String) {\n  throw UninitializedPropertyAccessException(\"lateinit property ${propertyName} has not been
initialized\")\n}\n\",/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n
*\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\npackage kotlin.collections\n\n/**\n * Groups elements from the [Grouping] source by key and counts elements
in each group.\n *\n * @return a [Map] associating the key of each group with the count of elements in the group.\n
*\n * @sample samples.collections.Grouping.groupingByEachCount\n *\n\n@SinceKotlin(\"1.1\")\npublic actual fun
<T, K> Grouping<T, K>.eachCount(): Map<K, Int> =\n  fold(0) { acc, _ -> acc + 1 }\n\n/**\n * Groups
elements from the [Grouping] source by key and sums values provided by the [valueSelector] function for elements
in each group.\n *\n * @return a [Map] associating the key of each group with the count of element in the group.\n
*\n\n@SinceKotlin(\"1.1\")\npublic inline fun <T, K> Grouping<T, K>.eachSumOf(valueSelector: (T) -> Int):
Map<K, Int> =\n  fold(0) { acc, e -> acc + valueSelector(e) }\n\n\",/*\n * Copyright 2010-2018 JetBrains s.r.o.
and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n
*\n\n\n@file:kotlin.jvm.JvmName(\"GroupingKt\")\n@file:kotlin.jvm.JvmMultifileClass\n\npackage
kotlin.collections\n\n/**\n * Represents a source of elements with a [keyOf] function, which can be applied to each
element to get its key.\n *\n * A [Grouping] structure serves as an intermediate step in group-and-fold operations:\n
*\n * they group elements by their keys and then fold each group with some aggregating operation.\n *\n * It is created
by attaching `keySelector: (T) -> K` function to a source of elements.\n * To get an instance of [Grouping] use one
of `groupingBy` extension functions:\n * - [Iterable.groupingBy]\n * - [Sequence.groupingBy]\n * -
[Array.groupingBy]\n * - [CharSequence.groupingBy]\n *\n * For the list of group-and-fold operations available,
see the [extension functions](#extension-functions) for `Grouping`.\n *\n\n@SinceKotlin(\"1.1\")\npublic interface
Grouping<T, out K> {\n  /** Returns an [Iterator] over the elements of the source of this grouping. *\n  fun
sourceIterator(): Iterator<T>\n  /** Extracts the key of an [element]. *\n  fun keyOf(element: T): K\n}\n\n/**\n *
Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group

```

sequentially, passing the previously accumulated value and the current element as arguments, and stores the results in a new map. The key for each element is provided by the [Grouping.keyOf] function. @param operation function is invoked on each element with the following parameters: - `key`: the key of the group this element belongs to; - `accumulator`: the current value of the accumulator of the group, can be `null` if it's the first `element` encountered in the group; - `element`: the element from the source being aggregated; - `first`: indicates whether it's the first `element` encountered in the group. @return a [Map] associating the key of each group with the result of aggregation of the group elements. @sample

```

samples.collections.Grouping.aggregateByRadix
*\n@SinceKotlin("1.1")\npublic inline fun <T, K, R>
Grouping<T, K>.aggregate(\n    operation: (key: K, accumulator: R?, element: T, first: Boolean) -> R)\n): Map<K,
R> {\n    return aggregateTo(mutableMapOf<K, R>(), operation)\n}\n\n**\n * Groups elements from the
[Grouping] source by key and applies [operation] to the elements of each group sequentially, passing the
previously accumulated value and the current element as arguments, and stores the results in the given
[destination] map. The key for each element is provided by the [Grouping.keyOf] function. @param
operation a function that is invoked on each element with the following parameters: - `key`: the key of the group
this element belongs to; - `accumulator`: the current value of the accumulator of the group, can be `null` if it's
the first `element` encountered in the group; - `element`: the element from the source being aggregated; -
`first`: indicates whether it's the first `element` encountered in the group. If the [destination] map already has
a value corresponding to some key, then the elements being aggregated for that key are never considered as
`first`. @return the [destination] map associating the key of each group with the result of aggregation of the
group elements. @sample samples.collections.Grouping.aggregateByRadixTo
*\n@SinceKotlin("1.1")\npublic inline fun <T, K, R, M : MutableMap<in K, R>> Grouping<T,
K>.aggregateTo(\n    destination: M,\n    operation: (key: K, accumulator: R?, element: T, first: Boolean) -> R)\n): M
{\n    for (e in this.sourceIterator()) {\n        val key = keyOf(e)\n        val accumulator = destination[key]\n
destination[key] = operation(key, accumulator, e, accumulator == null && !destination.containsKey(key))\n    }\n
return destination\n}\n\n**\n * Groups elements from the [Grouping] source by key and applies [operation] to the
elements of each group sequentially, passing the previously accumulated value and the current element as
arguments, and stores the results in a new map. An initial value of accumulator is provided by
[initialValueSelector] function. @param initialValueSelector a function that provides an initial value of
accumulator for each group. It's invoked with parameters: - `key`: the key of the group; - `element`: the
first element being encountered in that group. @param operation a function that is invoked on each element
with the following parameters: - `key`: the key of the group this element belongs to; - `accumulator`: the
current value of the accumulator of the group; - `element`: the element from the source being accumulated.
@return a [Map] associating the key of each group with the result of accumulating the group elements. @sample
samples.collections.Grouping.foldByEvenLengthWithComputedInitialValue
*\n@SinceKotlin("1.1")\npublic inline fun <T, K, R> Grouping<T, K>.fold(\n    initialValueSelector: (key: K,
element: T) -> R,\n    operation: (key: K, accumulator: R, element: T) -> R)\n): Map<K, R> =\n
@Suppress("UNCHECKED_CAST")\n    aggregate { key, acc, e, first -> operation(key, if (first)
initialValueSelector(key, e) else acc as R, e) }\n\n**\n * Groups elements from the [Grouping] source by key and
applies [operation] to the elements of each group sequentially, passing the previously accumulated value and the
current element as arguments, and stores the results in the given [destination] map. An initial value of
accumulator is provided by [initialValueSelector] function. @param initialValueSelector a function that
provides an initial value of accumulator for each group. It's invoked with parameters: - `key`: the key of the
group; - `element`: the first element being encountered in that group. If the [destination] map already has
a value corresponding to some key, that value is used as an initial value of the accumulator for that group and
the [initialValueSelector] function is not called for that group. @param operation a function that is invoked on
each element with the following parameters: - `key`: the key of the group this element belongs to; -
`accumulator`: the current value of the accumulator of the group; - `element`: the element from the source being
accumulated. @return the [destination] map associating the key of each group with the result of

```

```

accumulating the group elements.\n * @sample
samples.collections.Grouping.foldByEvenLengthWithComputedInitialValueTo\n *^\n@SinceKotlin("1.1")\npublic
inline fun <T, K, R, M : MutableMap<in K, R>> Grouping<T, K>.foldTo(\n destination: M,\n
initialValueSelector: (key: K, element: T) -> R,\n operation: (key: K, accumulator: R, element: T) -> R\n): M =\n
@Suppress("UNCHECKED_CAST")\n aggregateTo(destination) { key, acc, e, first -> operation(key, if (first)
initialValueSelector(key, e) else acc as R, e) }\n\n/**\n * Groups elements from the [Grouping] source by key and
applies [operation] to the elements of each group sequentially,\n * passing the previously accumulated value and the
current element as arguments, and stores the results in a new map.\n * An initial value of accumulator is the same
[initialValue] for each group.\n *\n * @param operation a function that is invoked on each element with the
following parameters:\n * - `accumulator`: the current value of the accumulator of the group;\n * - `element`: the
element from the source being accumulated.\n *\n * @return a [Map] associating the key of each group with the
result of accumulating the group elements.\n * @sample
samples.collections.Grouping.foldByEvenLengthWithConstantInitialValue\n *^\n@SinceKotlin("1.1")\npublic
inline fun <T, K, R> Grouping<T, K>.fold(\n initialValue: R,\n operation: (accumulator: R, element: T) -> R\n):
Map<K, R> =\n
@Suppress("UNCHECKED_CAST")\n aggregate { _, acc, e, first -> operation(if (first)
initialValue else acc as R, e) }\n\n/**\n * Groups elements from the [Grouping] source by key and applies
[operation] to the elements of each group sequentially,\n * passing the previously accumulated value and the current
element as arguments,\n * and stores the results in the given [destination] map.\n * An initial value of accumulator
is the same [initialValue] for each group.\n *\n * If the [destination] map already has a value corresponding to the
key of some group,\n * that value is used as an initial value of the accumulator for that group.\n *\n * @param operation
a function that is invoked on each element with the following parameters:\n * - `accumulator`: the current value of
the accumulator of the group;\n * - `element`: the element from the source being accumulated.\n *\n * @return the
[destination] map associating the key of each group with the result of accumulating the group elements.\n *
@sample samples.collections.Grouping.foldByEvenLengthWithConstantInitialValueTo\n
*^\n@SinceKotlin("1.1")\npublic inline fun <T, K, R, M : MutableMap<in K, R>> Grouping<T, K>.foldTo(\n
destination: M,\n initialValue: R,\n operation: (accumulator: R, element: T) -> R\n): M =\n
@Suppress("UNCHECKED_CAST")\n aggregateTo(destination) { _, acc, e, first -> operation(if (first)
initialValue else acc as R, e) }\n\n/**\n * Groups elements from the [Grouping] source by key and applies the
reducing [operation] to the elements of each group\n * sequentially starting from the second element of the group,\n
* passing the previously accumulated value and the current element as arguments,\n * and stores the results in a new
map.\n * An initial value of accumulator is the first element of the group.\n *\n * @param operation a function that
is invoked on each subsequent element of the group with the following parameters:\n * - `key`: the key of the group
this element belongs to;\n * - `accumulator`: the current value of the accumulator of the group;\n * - `element`: the
element from the source being accumulated.\n *\n * @return a [Map] associating the key of each group with the
result of accumulating the group elements.\n * @sample samples.collections.Grouping.reduceByMaxVowels\n
*^\n@SinceKotlin("1.1")\npublic inline fun <S, T : S, K> Grouping<T, K>.reduce(\n operation: (key: K,
accumulator: S, element: T) -> S\n): Map<K, S> =\n
aggregate { key, acc, e, first ->\n
@Suppress("UNCHECKED_CAST")\n if (first) e else operation(key, acc as S, e)\n }\n\n/**\n * Groups
elements from the [Grouping] source by key and applies the reducing [operation] to the elements of each group\n *
sequentially starting from the second element of the group,\n * passing the previously accumulated value and the
current element as arguments,\n * and stores the results in the given [destination] map.\n * An initial value of
accumulator is the first element of the group.\n *\n * If the [destination] map already has a value corresponding to
the key of some group,\n * that value is used as an initial value of the accumulator for that group and the first
element of that group is also\n * subjected to the [operation].\n *\n * @param operation a function that is invoked on
each subsequent element of the group with the following parameters:\n * - `accumulator`: the current value of the
accumulator of the group;\n * - `element`: the element from the source being folded;\n *\n * @return the
[destination] map associating the key of each group with the result of accumulating the group elements.\n *
@sample samples.collections.Grouping.reduceByMaxVowelsTo\n *^\n@SinceKotlin("1.1")\npublic inline fun <S,

```



```

String\n public fun stringify(o: Any?, replacer: ((key: String, value: Any?) -> Any?): String\n public fun
stringify(o: Any?, replacer: ((key: String, value: Any?) -> Any?)? = definedExternally, space: Int): String\n public
fun stringify(o: Any?, replacer: ((key: String, value: Any?) -> Any?)? = definedExternally, space: String): String\n
public fun stringify(o: Any?, replacer: Array<String>): String\n public fun stringify(o: Any?, replacer:
Array<String>, space: Int): String\n public fun stringify(o: Any?, replacer: Array<String>, space: String):
String\n\n public fun <T> parse(text: String): T\n public fun <T> parse(text: String, reviver: ((key: String, value:
Any?) -> Any?): T\n}\n", /*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\npackage kotlin.math\n\nimport kotlin.internal.InlineOnly\nimport kotlin.js.JsMath
as nativeMath\n\n// region ===== Double Math
=====
\n\n/** Computes the sine of the angle [x] given in
radians.\n * \n * Special cases:\n * - `sin(NaN|+Inf|-Inf)` is `NaN`\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun sin(x: Double): Double =
nativeMath.sin(x)\n\n/** Computes the cosine of the angle [x] given in radians.\n * \n * Special cases:\n * -
`cos(NaN|+Inf|-Inf)` is `NaN`\n * \n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun cos(x: Double):
Double = nativeMath.cos(x)\n\n/** Computes the tangent of the angle [x] given in radians.\n * \n * Special cases:\n
* - `tan(NaN|+Inf|-Inf)` is `NaN`\n * \n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun tan(x:
Double): Double = nativeMath.tan(x)\n\n/** Computes the arc sine of the value [x];\n * the returned value is an
angle in the range from `-PI/2` to `PI/2` radians.\n * \n * Special cases:\n * - `asin(x)` is `NaN`, when `abs(x) > 1`
or x is `NaN`\n * \n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun asin(x: Double): Double =
nativeMath.asin(x)\n\n/** Computes the arc cosine of the value [x];\n * the returned value is an angle in the
range from `0.0` to `PI` radians.\n * \n * Special cases:\n * - `acos(x)` is `NaN`, when `abs(x) > 1` or x is `NaN`\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun acos(x: Double): Double =
nativeMath.acos(x)\n\n/** Computes the arc tangent of the value [x];\n * the returned value is an angle in the
range from `-PI/2` to `PI/2` radians.\n * \n * Special cases:\n * - `atan(NaN)` is `NaN`\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun atan(x: Double): Double =
nativeMath.atan(x)\n\n/** Returns the angle `theta` of the polar coordinates `(r, theta)` that correspond\n * to the
rectangular coordinates `(x, y)` by computing the arc tangent of the value [y] / [x];\n * the returned value is an angle
in the range from `-PI` to `PI` radians.\n * \n * Special cases:\n * - `atan2(0.0, 0.0)` is `0.0`\n * - `atan2(0.0, x)` is
`0.0` for `x > 0` and `PI` for `x < 0`\n * - `atan2(-0.0, x)` is `-0.0` for `x > 0` and `-PI` for `x < 0`\n * - `atan2(y,
+Inf)` is `0.0` for `0 < y < +Inf` and `-0.0` for `-Inf < y < 0`\n * - `atan2(y, -Inf)` is `PI` for `0 < y < +Inf` and `-PI`
for `-Inf < y < 0`\n * - `atan2(y, 0.0)` is `PI/2` for `y > 0` and `-PI/2` for `y < 0`\n * - `atan2(+Inf, x)` is `PI/2` for
finite `x`\n * - `atan2(-Inf, x)` is `-PI/2` for finite `x`\n * - `atan2(NaN, x)` and `atan2(y, NaN)` is `NaN`\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun atan2(y: Double, x: Double): Double =
nativeMath.atan2(y, x)\n\n/** Computes the hyperbolic sine of the value [x].\n * \n * Special cases:\n * -
`sinh(NaN)` is `NaN`\n * - `sinh(+Inf)` is `+Inf`\n * - `sinh(-Inf)` is `-Inf`\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun sinh(x: Double): Double = nativeSinh(x)\n\n/**
Computes the hyperbolic cosine of the value [x].\n * \n * Special cases:\n * - `cosh(NaN)` is `NaN`\n * -
`cosh(+Inf|-Inf)` is `+Inf`\n * \n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun cosh(x: Double):
Double = nativeCosh(x)\n\n/** Computes the hyperbolic tangent of the value [x].\n * \n * Special cases:\n * -
`tanh(NaN)` is `NaN`\n * - `tanh(+Inf)` is `1.0`\n * - `tanh(-Inf)` is `-1.0`\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun tanh(x: Double): Double =
nativeTanh(x)\n\n/** Computes the inverse hyperbolic sine of the value [x].\n * \n * The returned value is `y`
such that `sinh(y) == x`.\n * \n * Special cases:\n * - `asinh(NaN)` is `NaN`\n * - `asinh(+Inf)` is `+Inf`\n * -
`asinh(-Inf)` is `-Inf`\n * \n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun asinh(x: Double): Double
= nativeAsinh(x)\n\n/** Computes the inverse hyperbolic cosine of the value [x].\n * \n * The returned value is
positive `y` such that `cosh(y) == x`.\n * \n * Special cases:\n * - `acosh(NaN)` is `NaN`\n * - `acosh(x)` is `NaN`
when `x < 1`\n * - `acosh(+Inf)` is `+Inf`\n * \n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun

```

`acosh(x: Double): Double = nativeAcosh(x)` Computes the inverse hyperbolic tangent of the value [x].
 The returned value is `y` such that $\tanh(y) == x$.
 Special cases:
 - $\tanh(\text{NaN})$ is NaN
 - $\tanh(x)$ is NaN when $x > 1$ or $x < -1$
 - $\tanh(1.0)$ is $+\text{Inf}$
 - $\tanh(-1.0)$ is $-\text{Inf}$

`@SinceKotlin("1.2")@InlineOnly public actual inline fun atanh(x: Double): Double = nativeAtanh(x)`
 Computes $\sqrt{x^2 + y^2}$ without intermediate overflow or underflow.
 Special cases:
 - returns $+\text{Inf}$ if any of arguments is infinite
 - returns NaN if any of arguments is NaN and the other is not infinite

`@SinceKotlin("1.2")@InlineOnly public actual inline fun hypot(x: Double, y: Double): Double = nativeHypot(x, y)`
 Computes the positive square root of the value [x].
 Special cases:
 - \sqrt{x} is NaN when $x < 0$ or x is NaN

`@SinceKotlin("1.2")@InlineOnly public actual inline fun sqrt(x: Double): Double = nativeMath.sqrt(x)`
 Computes Euler's number `e` raised to the power of the value [x].
 Special cases:
 - $\exp(\text{NaN})$ is NaN
 - $\exp(+\text{Inf})$ is $+\text{Inf}$
 - $\exp(-\text{Inf})$ is 0.0

`@SinceKotlin("1.2")@InlineOnly public actual inline fun exp(x: Double): Double = nativeMath.exp(x)`
 Computes $\exp(x) - 1$.
 This function can be implemented to produce more precise result for [x] near zero.
 Special cases:
 - $\expm1(\text{NaN})$ is NaN
 - $\expm1(+\text{Inf})$ is $+\text{Inf}$
 - $\expm1(-\text{Inf})$ is -1.0
 @see [exp] function.

`@SinceKotlin("1.2")@InlineOnly public actual inline fun expm1(x: Double): Double = nativeExpml(x)`
 Computes the logarithm of the value [x] to the given [base].
 Special cases:
 - $\log(x, b)$ is NaN if either `x` or `b` are NaN
 - $\log(x, b)$ is NaN when $x < 0$ or $b \leq 0$ or $b == 1.0$
 - $\log(+\text{Inf}, +\text{Inf})$ is NaN
 - $\log(+\text{Inf}, b)$ is $+\text{Inf}$ for $b > 1$ and $-\text{Inf}$ for $b < 1$
 - $\log(0.0, b)$ is $-\text{Inf}$ for $b > 1$ and $+\text{Inf}$ for $b > 1$
 See also logarithm functions for common fixed bases: [ln], [log10] and [log2].

`@SinceKotlin("1.2") public actual fun log(x: Double, base: Double): Double {`
 `if (base <= 0.0 || base == 1.0) return Double.NaN`
 `return nativeMath.log(x) / nativeMath.log(base)}`
 Computes the natural logarithm (base `E`) of the value [x].
 Special cases:
 - $\ln(\text{NaN})$ is NaN
 - $\ln(x)$ is NaN when $x < 0.0$
 - $\ln(+\text{Inf})$ is $+\text{Inf}$
 - $\ln(0.0)$ is $-\text{Inf}$

`@SinceKotlin("1.2")@InlineOnly public actual inline fun ln(x: Double): Double = nativeMath.log(x)`
 Computes the common logarithm (base 10) of the value [x].
 @see [ln] function for special cases.

`@SinceKotlin("1.2")@InlineOnly public actual inline fun log10(x: Double): Double = nativeLog10(x)`
 Computes the binary logarithm (base 2) of the value [x].
 @see [ln] function for special cases.

`@SinceKotlin("1.2")@InlineOnly public actual inline fun log2(x: Double): Double = nativeLog2(x)`
 Computes $\ln(x + 1)$.
 This function can be implemented to produce more precise result for [x] near zero.
 Special cases:
 - $\ln1p(\text{NaN})$ is NaN
 - $\ln1p(x)$ is NaN where $x < -1.0$
 - $\ln1p(-1.0)$ is $-\text{Inf}$
 - $\ln1p(+\text{Inf})$ is $+\text{Inf}$
 @see [ln] function
 @see [expm1] function

`@SinceKotlin("1.2")@InlineOnly public actual inline fun ln1p(x: Double): Double = nativeLog1p(x)`
 Rounds the given value [x] to an integer towards positive infinity.
 @return the smallest double value that is greater than or equal to the given value [x] and is a mathematical integer.
 Special cases:
 - $\text{ceil}(x)$ is x where x is NaN or $+\text{Inf}$ or $-\text{Inf}$ or already a mathematical integer.

`@SinceKotlin("1.2")@InlineOnly public actual inline fun ceil(x: Double): Double = nativeMath.ceil(x)`
 Rounds the given value [x] to an integer towards negative infinity.
 @return the largest double value that is smaller than or equal to the given value [x] and is a mathematical integer.
 Special cases:
 - $\text{floor}(x)$ is x where x is NaN or $+\text{Inf}$ or $-\text{Inf}$ or already a mathematical integer.

`@SinceKotlin("1.2")@InlineOnly public actual inline fun floor(x: Double): Double = nativeMath.floor(x)`
 Rounds the given value [x] to an integer towards zero.
 @return the value [x] having its fractional part truncated.
 Special cases:
 - $\text{truncate}(x)$ is x where x is NaN or $+\text{Inf}$ or $-\text{Inf}$ or already a mathematical integer.

`@SinceKotlin("1.2")@InlineOnly public actual inline fun truncate(x: Double): Double = nativeTrunc(x)`
 Rounds the given value [x] towards the closest integer with ties rounded towards even integer.
 Special cases:
 - $\text{round}(x)$ is x where x is NaN or $+\text{Inf}$ or $-\text{Inf}$ or already a mathematical integer.

`@SinceKotlin("1.2") public actual fun round(x: Double): Double {`
 `if (x % 0.5 != 0.0) {`
 `return nativeMath.round(x)`
 `} val floor = floor(x)`
 `return if (floor % 2 == 0.0) floor else ceil(x)}`
 Returns the absolute value of the given value [x].
 Special cases:
 -

```

`abs(NaN)` is `NaN` \n * @see absoluteValue extension property for [Double] \n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun abs(x: Double): Double =
nativeMath.abs(x)\n\n/** \n * Returns the sign of the given value [x]: \n * - `1.0` if the value is negative, \n * - zero
if the value is zero, \n * - `1.0` if the value is positive \n * \n * Special case: \n * - `sign(NaN)` is `NaN` \n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun sign(x: Double): Double =
nativeSign(x)\n\n/** \n * Returns the smaller of two values. \n * \n * If either value is `NaN`, then the result is
`NaN`. \n * \n *\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun min(a: Double, b: Double): Double =
nativeMath.min(a, b)\n\n/** \n * Returns the greater of two values. \n * \n * If either value is `NaN`, then the result is
`NaN`. \n * \n *\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun max(a: Double, b: Double): Double =
nativeMath.max(a, b)\n\n/** \n * Returns the cube root of [x]. For any `x`, `cbrt(-x) == -cbrt(x)`; \n * that is, the
cube root of a negative value is the negative of the cube root \n * of that value's magnitude. Special cases: \n * \n *
Special cases: \n * - If the argument is `NaN`, then the result is `NaN`. \n * - If the argument is infinite, then the
result is an infinity with the same sign as the argument. \n * - If the argument is zero, then the result is a zero with
the same sign as the argument. \n * \n *\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@InlineOnly\npublic actual
inline fun cbrt(x: Double): Double = nativeMath.cbrt(x)\n\n// extensions\n\n/** \n * Raises this value to the power
[x]. \n * \n * Special cases: \n * - `b.pow(0.0)` is `1.0` \n * - `b.pow(1.0) == b` \n * - `b.pow(NaN)` is `NaN` \n * -
`NaN.pow(x)` is `NaN` for `x != 0.0` \n * - `b.pow(Inf)` is `NaN` for `abs(b) == 1.0` \n * - `b.pow(x)` is `NaN` for
`b < 0` and `x` is finite and not an integer \n * \n *\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun
Double.pow(x: Double): Double = nativeMath.pow(this, x)\n\n/** \n * Raises this value to the integer power [n]. \n *
\n * See the other overload of [pow] for details. \n * \n *\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline
fun Double.pow(n: Int): Double = nativeMath.pow(this, n.toDouble())\n\n/** \n * Returns the absolute value of this
value. \n * \n * Special cases: \n * - `NaN.absoluteValue` is `NaN` \n * \n * @see abs function \n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline val Double.absoluteValue: Double get() =
nativeMath.abs(this)\n\n/** \n * Returns the sign of this value: \n * - `1.0` if the value is negative, \n * - zero if the
value is zero, \n * - `1.0` if the value is positive \n * \n * Special case: \n * - `NaN.sign` is `NaN` \n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline val Double.sign: Double get() =
nativeSign(this)\n\n/** \n * Returns this value with the sign bit same as of the [sign] value. \n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun Double.withSign(sign: Int): Double =
this.withSign(sign.toDouble())\n\n/** \n * Returns the ulp (unit in the last place) of this value. \n * \n * An ulp is a
positive distance between this value and the next nearest [Double] value larger in magnitude. \n * \n * Special
Cases: \n * - `NaN.ulp` is `NaN` \n * - `x.ulp` is `+Inf` when `x` is `+Inf` or `-Inf` \n * - `0.0.ulp` is
`Double.MIN_VALUE` \n * \n *\n@SinceKotlin("1.2")\npublic actual val Double.ulp: Double get() = when {\n    this
< 0 -> (-this).ulp\n    this.isNaN() || this == Double.POSITIVE_INFINITY -> this\n    this ==
Double.MAX_VALUE -> this - this.nextDown()\n    else -> this.nextUp() - this\n}\n\n/** \n * Returns the [Double]
value nearest to this value in direction of positive infinity. \n * \n *\n@SinceKotlin("1.2")\npublic actual fun
Double.nextUp(): Double = when {\n    this.isNaN() || this == Double.POSITIVE_INFINITY -> this\n    this == 0.0
-> Double.MIN_VALUE\n    else -> Double.fromBits(this.toRawBits() + if (this > 0) 1 else -1)\n}\n\n/** \n *
Returns the [Double] value nearest to this value in direction of negative infinity. \n
*\n@SinceKotlin("1.2")\npublic actual fun Double.nextDown(): Double = when {\n    this.isNaN() || this ==
Double.NEGATIVE_INFINITY -> this\n    this == 0.0 -> -Double.MIN_VALUE\n    else ->
Double.fromBits(this.toRawBits() + if (this > 0) -1 else 1)\n}\n\n/** \n * Returns the [Double] value nearest to this
value in direction from this value towards the value [to]. \n * \n * Special cases: \n * - `x.nextTowards(y)` is `NaN` if
either `x` or `y` are `NaN` \n * - `x.nextTowards(x) == x` \n * \n *\n@SinceKotlin("1.2")\npublic actual fun
Double.nextTowards(to: Double): Double = when {\n    this.isNaN() || to.isNaN() -> Double.NaN\n    to == this ->
to\n    to > this -> this.nextUp()\n    else /* to < this */ -> this.nextDown()\n}\n\n/** \n * Rounds this [Double]
value to the nearest integer and converts the result to [Int]. \n * \n * Ties are rounded towards positive infinity. \n *
\n * Special cases: \n * - `x.roundToInt() == Int.MAX_VALUE` when `x > Int.MAX_VALUE` \n * - `x.roundToInt()
== Int.MIN_VALUE` when `x < Int.MIN_VALUE` \n * \n * @throws IllegalArgumentException when this value is

```

```

`NaN` *  

@SinceKotlin("1.2")  

public actual fun Double.roundToInt(): Int = when {  

    isNaN() -> throw  

    IllegalArgumentException("Cannot round NaN value.")  

    this > Int.MAX_VALUE -> Int.MAX_VALUE  

    this < Int.MIN_VALUE -> Int.MIN_VALUE  

    else -> nativeMath.round(this).toInt()  

} *  

Rounds this  

[Double] value to the nearest integer and converts the result to [Long].  

*  

Ties are rounded towards positive  

infinity.  

*  

Special cases:  

- `x.roundToLong() == Long.MAX_VALUE` when `x >  

Long.MAX_VALUE`  

- `x.roundToLong() == Long.MIN_VALUE` when `x < Long.MIN_VALUE`  

*  

@throws IllegalArgumentException when this value is `NaN` *  

@SinceKotlin("1.2")  

public actual fun  

Double.roundToLong(): Long = when {  

    isNaN() -> throw IllegalArgumentException("Cannot round NaN  

value.")  

    this > Long.MAX_VALUE -> Long.MAX_VALUE  

    this < Long.MIN_VALUE ->  

Long.MIN_VALUE  

    else -> nativeMath.round(this).toLong()  

} // endregion  

// region  

===== Float Math =====  

*  

Computes the  

sine of the angle [x] given in radians.  

*  

Special cases:  

- `sin(NaN|+Inf|-Inf)` is `NaN`  

*  

@SinceKotlin("1.2")  

@InlineOnly  

public actual inline fun sin(x: Float): Float =  

nativeMath.sin(x.toDouble()).toFloat()  

*  

Computes the cosine of the angle [x] given in radians.  

*  

Special  

cases:  

- `cos(NaN|+Inf|-Inf)` is `NaN`  

*  

@SinceKotlin("1.2")  

@InlineOnly  

public actual inline fun  

cos(x: Float): Float = nativeMath.cos(x.toDouble()).toFloat()  

*  

Computes the tangent of the angle [x] given in  

radians.  

*  

Special cases:  

- `tan(NaN|+Inf|-Inf)` is `NaN`  

*  

@SinceKotlin("1.2")  

@InlineOnly  

public actual inline fun tan(x: Float): Float =  

nativeMath.tan(x.toDouble()).toFloat()  

*  

Computes the arc sine of the value [x];  

*  

the returned value is  

an angle in the range from  $-\pi/2$  to  $\pi/2$  radians.  

*  

Special cases:  

- `asin(x)` is `NaN`, when `abs(x) >  

1` or x is `NaN`  

*  

@SinceKotlin("1.2")  

@InlineOnly  

public actual inline fun asin(x: Float): Float =  

nativeMath.asin(x.toDouble()).toFloat()  

*  

Computes the arc cosine of the value [x];  

*  

the returned value  

is an angle in the range from  $0.0$  to  $\pi$  radians.  

*  

Special cases:  

- `acos(x)` is `NaN`, when `abs(x) >  

1` or x is `NaN`  

*  

@SinceKotlin("1.2")  

@InlineOnly  

public actual inline fun acos(x: Float): Float =  

nativeMath.acos(x.toDouble()).toFloat()  

*  

Computes the arc tangent of the value [x];  

*  

the returned value  

is an angle in the range from  $-\pi/2$  to  $\pi/2$  radians.  

*  

Special cases:  

- `atan(NaN)` is `NaN`  

*  

@SinceKotlin("1.2")  

@InlineOnly  

public actual inline fun atan(x: Float): Float =  

nativeMath.atan(x.toDouble()).toFloat()  

*  

Returns the angle `theta` of the polar coordinates `(r, theta)` that  

correspond  

*  

to the rectangular coordinates `(x, y)` by computing the arc tangent of the value  $y / x$ ;  

*  

the  

returned value is an angle in the range from  $-\pi$  to  $\pi$  radians.  

*  

Special cases:  

- `atan2(0.0, 0.0)` is  

 $0.0$   

- `atan2(0.0, x)` is  $0.0$  for  $x > 0$  and  $\pi$  for  $x < 0$   

- `atan2(-0.0, x)` is  $-0.0$  for  $x > 0$  and  $-\pi$   

for  $x < 0$   

- `atan2(y, +Inf)` is  $0.0$  for  $0 < y < +Inf$  and  $-0.0$  for  $-\infty < y < 0$   

- `atan2(y, -Inf)` is  $\pi$   

for  $0 < y < +Inf$  and  $-\pi$  for  $-\infty < y < 0$   

- `atan2(y, 0.0)` is  $\pi/2$  for  $y > 0$  and  $-\pi/2$  for  $y < 0$   

-  

`atan2(+Inf, x)` is  $\pi/2$  for finite  $x \neq 0$   

- `atan2(-Inf, x)` is  $-\pi/2$  for finite  $x \neq 0$   

- `atan2(NaN, x)` and  

`atan2(y, NaN)` is `NaN`  

*  

@SinceKotlin("1.2")  

@InlineOnly  

public actual inline fun atan2(y: Float, x:  

Float): Float = nativeMath.atan2(y.toDouble(), x.toDouble()).toFloat()  

*  

Computes the hyperbolic sine of  

the value [x].  

*  

Special cases:  

- `sinh(NaN)` is `NaN`  

- `sinh(+Inf)` is  $+\infty$   

- `sinh(-Inf)` is  $-\infty$   

*  

@SinceKotlin("1.2")  

@InlineOnly  

public actual inline fun sinh(x: Float): Float =  

nativeSinh(x.toDouble()).toFloat()  

*  

Computes the hyperbolic cosine of the value [x].  

*  

Special  

cases:  

- `cosh(NaN)` is `NaN`  

- `cosh(+Inf|-Inf)` is  $+\infty$   

*  

@SinceKotlin("1.2")  

@InlineOnly  

public actual inline fun cosh(x: Float): Float =  

nativeCosh(x.toDouble()).toFloat()  

*  

Computes the hyperbolic tangent of the value [x].  

*  

Special  

cases:  

- `tanh(NaN)` is `NaN`  

- `tanh(+Inf)` is  $1.0$   

- `tanh(-Inf)` is  $-1.0$   

*  

@SinceKotlin("1.2")  

@InlineOnly  

public actual inline fun tanh(x: Float): Float =  

nativeTanh(x.toDouble()).toFloat()  

*  

Computes the inverse hyperbolic sine of the value [x].  

*  

The  

returned value is `y` such that  $\sinh(y) == x$ .  

*  

Special cases:  

- `asinh(NaN)` is `NaN`  

-  

`asinh(+Inf)` is  $+\infty$   

- `asinh(-Inf)` is  $-\infty$   

*  

@SinceKotlin("1.2")  

@InlineOnly  

public actual inline  

fun asinh(x: Float): Float = nativeAsinh(x.toDouble()).toFloat()  

*  

Computes the inverse hyperbolic cosine

```

of the value $[x]$.
 The returned value is positive y such that $\cosh(y) == x$.
 Special cases:
 $\operatorname{acosh}(\text{NaN})$ is NaN
 $\operatorname{acosh}(x)$ is NaN when $x < 1$
 $\operatorname{acosh}(+\infty)$ is $+\infty$

```

*\/n@SinceKotlin("1.2")n@InlineOnlynpublic actual inline fun acosh(x: Float): Float =
nativeAcosh(x.toDouble()).toFloat()n/n/**n * Computes the inverse hyperbolic tangent of the value [x].n *n *
The returned value is `y` such that `tanh(y) == x`.n *n * Special cases:n * - `tanh(NaN)` is `NaN`n * - `tanh(x)`
is `NaN` when `x > 1` or `x < -1`n * - `tanh(1.0)` is `+Inf`n * - `tanh(-1.0)` is `-Inf`n
*/n@SinceKotlin("1.2")n@InlineOnlynpublic actual inline fun atanh(x: Float): Float =
nativeAtanh(x.toDouble()).toFloat()n/n/**n * Computes `sqrt(x^2 + y^2)` without intermediate overflow or
underflow.n *n * Special cases:n * - returns `+Inf` if any of arguments is infiniten * - returns `NaN` if any of
arguments is `NaN` and the other is not infiniten *\/n@SinceKotlin("1.2")n@InlineOnlynpublic actual inline fun
hypot(x: Float, y: Float): Float = nativeHypot(x.toDouble(), y.toDouble()).toFloat()n/n/**n * Computes the
positive square root of the value [x].n *n * Special cases:n * - `sqrt(x)` is `NaN` when `x < 0` or `x` is `NaN`n
*/n@SinceKotlin("1.2")n@InlineOnlynpublic actual inline fun sqrt(x: Float): Float =
nativeMath.sqrt(x.toDouble()).toFloat()n/n/**n * Computes Euler's number `e` raised to the power of the value
[x].n *n * Special cases:n * - `exp(NaN)` is `NaN`n * - `exp(+Inf)` is `+Inf`n * - `exp(-Inf)` is `0.0`n
*/n@SinceKotlin("1.2")n@InlineOnlynpublic actual inline fun exp(x: Float): Float =
nativeMath.exp(x.toDouble()).toFloat()n/n/**n * Computes `exp(x) - 1`.n *n * This function can be implemented
to produce more precise result for [x] near zero.n *n * Special cases:n * - `expm1(NaN)` is `NaN`n * -
`expm1(+Inf)` is `+Inf`n * - `expm1(-Inf)` is `-1.0`n *n * @see [exp] function.n
*/n@SinceKotlin("1.2")n@InlineOnlynpublic actual inline fun expm1(x: Float): Float =
nativeExpml(x.toDouble()).toFloat()n/n/**n * Computes the logarithm of the value [x] to the given [base].n *n *
Special cases:n * - `log(x, b)` is `NaN` if either `x` or `b` are `NaN`n * - `log(x, b)` is `NaN` when `x < 0` or `b
<= 0` or `b == 1.0`n * - `log(+Inf, +Inf)` is `NaN`n * - `log(+Inf, b)` is `+Inf` for `b > 1` and `-Inf` for `b < 1`n
* - `log(0.0, b)` is `-Inf` for `b > 1` and `+Inf` for `b > 1`n *n * See also logarithm functions for common fixed
bases: [ln], [log10] and [log2].n *\/n@SinceKotlin("1.2")n@InlineOnlynpublic actual inline fun log(x: Float,
base: Float): Float = log(x.toDouble(), base.toDouble()).toFloat()n/n/**n * Computes the natural logarithm (base
`E`) of the value [x].n *n * Special cases:n * - `ln(NaN)` is `NaN`n * - `ln(x)` is `NaN` when `x < 0.0`n * -
`ln(+Inf)` is `+Inf`n * - `ln(0.0)` is `-Inf`n *\/n@SinceKotlin("1.2")n@InlineOnlynpublic actual inline fun ln(x:
Float): Float = nativeMath.log(x.toDouble()).toFloat()n/n/**n * Computes the common logarithm (base 10) of the
value [x].n *n * @see [ln] function for special cases.n *\/n@SinceKotlin("1.2")n@InlineOnlynpublic actual
inline fun log10(x: Float): Float = nativeLog10(x.toDouble()).toFloat()n/n/**n * Computes the binary logarithm
(base 2) of the value [x].n *n * @see [ln] function for special cases.n
*/n@SinceKotlin("1.2")n@InlineOnlynpublic actual inline fun log2(x: Float): Float =
nativeLog2(x.toDouble()).toFloat()n/n/**n * Computes `ln(a + 1)`.n *n * This function can be implemented to
produce more precise result for [x] near zero.n *n * Special cases:n * - `ln1p(NaN)` is `NaN`n * - `ln1p(x)` is
`NaN` where `x < -1.0`n * - `ln1p(-1.0)` is `-Inf`n * - `ln1p(+Inf)` is `+Inf`n *n * @see [ln] functionn * @see
[expm1] functionn *\/n@SinceKotlin("1.2")n@InlineOnlynpublic actual inline fun ln1p(x: Float): Float =
nativeLog1p(x.toDouble()).toFloat()n/n/**n * Rounds the given value [x] to an integer towards positive
infinity.n *n * @return the smallest Float value that is greater than or equal to the given value [x] and is a
mathematical integer.n *n * Special cases:n * - `ceil(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a
mathematical integer.n *\/n@SinceKotlin("1.2")n@InlineOnlynpublic actual inline fun ceil(x: Float): Float =
nativeMath.ceil(x.toDouble()).toFloat()n/n/**n * Rounds the given value [x] to an integer towards negative
infinity.n *n * @return the largest Float value that is smaller than or equal to the given value [x] and is a
mathematical integer.n *n * Special cases:n * - `floor(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a
mathematical integer.n *\/n@SinceKotlin("1.2")n@InlineOnlynpublic actual inline fun floor(x: Float): Float =
nativeMath.floor(x.toDouble()).toFloat()n/n/**n * Rounds the given value [x] to an integer towards zero.n *n *
@return the value [x] having its fractional part truncated.n *n * Special cases:n * - `truncate(x)` is `x` where `x`
is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.n *\/n@SinceKotlin("1.2")n@InlineOnlynpublic

```

actual inline fun truncate(x: Float): Float = truncate(x.toDouble()).toFloat()\n\n**\n * Rounds the given value [x] towards the closest integer with ties rounded towards even integer.\n * \n * Special cases:\n * - `round(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n

*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun round(x: Float): Float = round(x.toDouble()).toFloat()\n\n**\n * Returns the absolute value of the given value [x].\n * \n * Special cases:\n * - `abs(NaN)` is `NaN`\n * \n * @see absoluteValue extension property for [Float]\n

*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun abs(x: Float): Float = nativeMath.abs(x.toDouble()).toFloat()\n\n**\n * Returns the sign of the given value [x]:\n * - `-1.0` if the value is negative,\n * - zero if the value is zero,\n * - `1.0` if the value is positive\n * \n * Special case:\n * - `sign(NaN)` is `NaN`\n * \n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun sign(x: Float): Float = nativeSign(x.toDouble()).toFloat()\n\n**\n * Returns the smaller of two values.\n * \n * If either value is `NaN`, then the result is `NaN`.\n * \n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun min(a: Float, b: Float): Float = nativeMath.min(a, b)\n\n**\n * Returns the greater of two values.\n * \n * If either value is `NaN`, then the result is `NaN`.\n * \n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun max(a: Float, b: Float): Float = nativeMath.max(a, b)\n\n**\n * Returns the cube root of [x]. For any `x`, `cbrt(-x) == -cbrt(x)`; \n * that is, the cube root of a negative value is the negative of the cube root\n * of that value's magnitude. Special cases:\n * \n * Special cases:\n * - If the argument is `NaN`, then the result is `NaN`.\n * - If the argument is infinite, then the result is an infinity with the same sign as the argument.\n * - If the argument is zero, then the result is a zero with the same sign as the argument.\n * \n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@InlineOnly\npublic actual inline fun cbrt(x: Float): Float = nativeMath.cbrt(x.toDouble()).toFloat()\n\n// extensions\n\n**\n * Raises this value to the power [x].\n * \n * Special cases:\n * - `b.pow(0.0)` is `1.0`\n * - `b.pow(1.0) == b`\n * - `b.pow(NaN)` is `NaN`\n * - `NaN.pow(x)` is `NaN` for `x != 0.0`\n * - `b.pow(Inf)` is `NaN` for `abs(b) == 1.0`\n * - `b.pow(x)` is `NaN` for `b < 0` and `x` is finite and not an integer\n

*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun Float.pow(x: Float): Float = nativeMath.pow(this.toDouble(), x.toDouble()).toFloat()\n\n**\n * Raises this value to the integer power [n].\n * \n * See the other overload of [pow] for details.\n * \n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun Float.pow(n: Int): Float = nativeMath.pow(this.toDouble(), n.toDouble()).toFloat()\n\n**\n * Returns the absolute value of this value.\n * \n * Special cases:\n * - `NaN.absoluteValue` is `NaN`\n * \n * @see abs function\n

*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline val Float.absoluteValue: Float get() = nativeMath.abs(this.toDouble()).toFloat()\n\n**\n * Returns the sign of this value:\n * - `-1.0` if the value is negative,\n * - zero if the value is zero,\n * - `1.0` if the value is positive\n * \n * Special case:\n * - `NaN.sign` is `NaN`\n * \n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline val Float.sign: Float get() = nativeSign(this.toDouble()).toFloat()\n\n**\n * Returns this value with the sign bit same as of the [sign] value.\n * \n * If [sign] is `NaN` the sign of the result is undefined.\n * \n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun Float.withSign(sign: Float): Float = this.toDouble().withSign(sign.toDouble()).toFloat()\n\n**\n * Returns this value with the sign bit same as of the [sign] value.\n * \n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun Float.withSign(sign: Int): Float = this.toDouble().withSign(sign.toDouble()).toFloat()\n\n**\n * Rounds this [Float] value to the nearest integer and converts the result to [Int].\n * Ties are rounded towards positive infinity.\n * \n * Special cases:\n * - `x.roundToInt() == Int.MAX_VALUE` when `x > Int.MAX_VALUE`\n * - `x.roundToInt() == Int.MIN_VALUE` when `x < Int.MIN_VALUE`\n * \n * @throws IllegalArgumentException when this value is `NaN`\n * \n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun Float.roundToInt(): Int = toDouble().roundToInt()\n\n**\n * Rounds this [Float] value to the nearest integer and converts the result to [Long].\n * Ties are rounded towards positive infinity.\n * \n * Special cases:\n * - `x.roundToLong() == Long.MAX_VALUE` when `x > Long.MAX_VALUE`\n * - `x.roundToLong() == Long.MIN_VALUE` when `x < Long.MIN_VALUE`\n * \n * @throws IllegalArgumentException when this value is `NaN`\n

*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun Float.roundToLong(): Long = toDouble().roundToLong()\n\n// endregion\n\n// region ===== Integer Math =====\n\n**\n * Returns the absolute value of the given value

```

[n].n * Special cases: n * - `abs(Int.MIN_VALUE)` is `Int.MIN_VALUE` due to an overflow n * @see
absoluteValue extension property for [Int] n * // TODO: remove manual 'or' when KT-19290 is
fixed n @SinceKotlin("1.2") n public actual fun abs(n: Int): Int = if (n < 0) (-n or 0) else n n / n * Returns the
smaller of two values. n * n @SinceKotlin("1.2") n @InlineOnly n public actual inline fun min(a: Int, b: Int): Int =
nativeMath.min(a, b) n / n * Returns the greater of two values. n
* n @SinceKotlin("1.2") n @InlineOnly n public actual inline fun max(a: Int, b: Int): Int = nativeMath.max(a,
b) n / n * Returns the absolute value of this value. n * n * Special cases: n * -
`Int.MIN_VALUE.absoluteValue` is `Int.MIN_VALUE` due to an overflow n * n * @see abs function n
* n @SinceKotlin("1.2") n @InlineOnly n public actual inline val Int.absoluteValue: Int get() = abs(this) n / n * n *
Returns the sign of this value: n * - -1` if the value is negative, n * - `0` if the value is zero, n * - `1` if the value
is positive n * n @SinceKotlin("1.2") n public actual val Int.sign: Int get() = when { n this < 0 -> -1 n this > 0 -
> 1 n else -> 0 } n / n / n * Returns the absolute value of the given value [n]. n * n * Special cases: n * -
`abs(Long.MIN_VALUE)` is `Long.MIN_VALUE` due to an overflow n * n * @see absoluteValue extension
property for [Long] n * n @SinceKotlin("1.2") n public actual fun abs(n: Long): Long = if (n < 0) -n else
n n / n * Returns the smaller of two values. n
* n @SinceKotlin("1.2") n @Suppress("NOTHING_TO_INLINE") n public actual inline fun min(a: Long, b:
Long): Long = if (a <= b) a else b n / n * Returns the greater of two values. n
* n @SinceKotlin("1.2") n @Suppress("NOTHING_TO_INLINE") n public actual inline fun max(a: Long, b:
Long): Long = if (a >= b) a else b n / n * Returns the absolute value of this value. n * n * Special cases: n * -
`Long.MIN_VALUE.absoluteValue` is `Long.MIN_VALUE` due to an overflow n * n * @see abs function n
* n @SinceKotlin("1.2") n @InlineOnly n public actual inline val Long.absoluteValue: Long get() =
abs(this) n / n * Returns the sign of this value: n * - -1` if the value is negative, n * - `0` if the value is zero, n
* - `1` if the value is positive n * n @SinceKotlin("1.2") n public actual val Long.sign: Int get() = when { n this
< 0 -> -1 n this > 0 -> 1 n else -> 0 } n / n / n // endregion n", /* n * Copyright 2010-2021 JetBrains s.r.o. and
Kotlin Programming Language contributors. n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file. n * n package kotlin n / n * Returns `true` if the specified
number is a n * Not-a-Number (NaN) value, `false` otherwise. n * n public actual fun Double.isNaN(): Boolean =
this != this n / n * Returns `true` if the specified number is a n * Not-a-Number (NaN) value, `false` otherwise. n
* n public actual fun Float.isNaN(): Boolean = this != this n / n * Returns `true` if this value is infinitely large in
magnitude. n * n public actual fun Double.isInfinite(): Boolean = this == Double.POSITIVE_INFINITY || this ==
Double.NEGATIVE_INFINITY n / n * Returns `true` if this value is infinitely large in magnitude. n * n public
actual fun Float.isInfinite(): Boolean = this == Float.POSITIVE_INFINITY || this ==
Float.NEGATIVE_INFINITY n / n * Returns `true` if the argument is a finite floating-point value; returns
`false` otherwise (for `NaN` and infinity arguments). n * n public actual fun Double.isFinite(): Boolean =
!isInfinite() && !isNaN() n / n * Returns `true` if the argument is a finite floating-point value; returns `false`
otherwise (for `NaN` and infinity arguments). n * n public actual fun Float.isFinite(): Boolean = !isInfinite() &&
!isNaN() n / n * Counts the number of set bits in the binary representation of this [Int] number. n
* n @SinceKotlin("1.4") n @WasExperimental(ExperimentalStdlibApi::class) n public actual fun
Int.countOneBits(): Int { n // Hacker's Delight 5-1 algorithm n var v = this n v = (v and 0x55555555) +
(v.ushr(1) and 0x55555555) n v = (v and 0x33333333) + (v.ushr(2) and 0x33333333) n v = (v and 0x0F0F0F0F)
+ (v.ushr(4) and 0x0F0F0F0F) n v = (v and 0x00FF00FF) + (v.ushr(8) and 0x00FF00FF) n v = (v and
0x0000FFFF) + (v.ushr(16)) n return v } n / n * Counts the number of consecutive most significant bits that
are zero in the binary representation of this [Int] number. n
* n @SinceKotlin("1.4") n @WasExperimental(ExperimentalStdlibApi::class) n @kotlin.internal.InlineOnly n publi
c actual inline fun Int.countLeadingZeroBits(): Int = nativeClz32(this) n / n * Counts the number of consecutive
least significant bits that are zero in the binary representation of this [Int] number. n
* n @SinceKotlin("1.4") n @WasExperimental(ExperimentalStdlibApi::class) n public actual fun
Int.countTrailingZeroBits(): Int = n // Hacker's Delight 5-4 algorithm for expressing countTrailingZeroBits with

```

```

countLeadingZeroBits\n  Int.SIZE_BITS - (this or -this).inv().countLeadingZeroBits()\n\n/**\n * Returns a
number having a single bit set in the position of the most significant set bit of this [Int] number,\n * or zero, if this
number is zero.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Int.takeHighestOneBit(): Int =\n  if (this == 0) 0 else 1.shl(Int.SIZE_BITS - 1 - countLeadingZeroBits())\n\n/**\n *
Returns a number having a single bit set in the position of the least significant set bit of this [Int] number,\n * or
zero, if this number is zero.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Int.takeLowestOneBit(): Int =\n  // Hacker's Delight 2-1 algorithm for isolating rightmost 1-bit\n  this and -
this\n\n/**\n * Rotates the binary representation of this [Int] number left by the specified [bitCount] number of
bits.\n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the
right side.\n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit
count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n * Rotating by a multiple of [Int.SIZE_BITS] (32)
returns the same number, or more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 32)`\n */\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Int.rotateLeft(bitCount: Int): Int =\n  shl(bitCount) or ushr(Int.SIZE_BITS - bitCount)\n\n/**\n * Rotates the
binary representation of this [Int] number right by the specified [bitCount] number of bits.\n * The least significant
bits pushed out from the right side reenter the number as the most significant bits on the left side.\n * Rotating
the number right by a negative bit count is the same as rotating it left by the negated bit count:\n *
`number.rotateRight(-n) == number.rotateLeft(n)`\n * Rotating by a multiple of [Int.SIZE_BITS] (32) returns
the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 32)`\n */\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Int.rotateRight(bitCount: Int): Int =\n  shl(Int.SIZE_BITS - bitCount) or ushr(bitCount)\n\n/**\n * Counts the
number of set bits in the binary representation of this [Long] number.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Long.countOneBits(): Int =\n  high.countOneBits() + low.countOneBits()\n\n/**\n * Counts the number of
consecutive most significant bits that are zero in the binary representation of this [Long] number.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Long.countLeadingZeroBits(): Int =\n  when (val high = this.high) {\n    0 -> Int.SIZE_BITS +
low.countLeadingZeroBits()\n    else -> high.countLeadingZeroBits()\n  }\n\n/**\n * Counts the number of
consecutive least significant bits that are zero in the binary representation of this [Long] number.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Long.countTrailingZeroBits(): Int =\n  when (val low = this.low) {\n    0 -> Int.SIZE_BITS +
high.countTrailingZeroBits()\n    else -> low.countTrailingZeroBits()\n  }\n\n/**\n * Returns a number having a
single bit set in the position of the most significant set bit of this [Long] number,\n * or zero, if this number is
zero.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Long.takeHighestOneBit(): Long =\n  when (val high = this.high) {\n    0 -> Long(low.takeHighestOneBit(),
0)\n    else -> Long(0, high.takeHighestOneBit())\n  }\n\n/**\n * Returns a number having a single bit set in the
position of the least significant set bit of this [Long] number,\n * or zero, if this number is zero.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Long.takeLowestOneBit(): Long =\n  when (val low = this.low) {\n    0 -> Long(0, high.takeLowestOneBit())\n
else -> Long(low.takeLowestOneBit(), 0)\n  }\n\n/**\n * Rotates the binary representation of this [Long]
number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side
reenter the number as the least significant bits on the right side.\n * Rotating the number left by a negative bit
count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) ==
number.rotateRight(n)`\n * Rotating by a multiple of [Long.SIZE_BITS] (64) returns the same number, or more
generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 64)`\n */\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Long.rotateLeft(bitCount: Int): Long {\n  if ((bitCount and 31) != 0) {\n    val low = this.low\n    val high =

```



```

Annotation> KClass<*>.findAssociatedObject(): Any? =\n  this.findAssociatedObject(T::class)", /*\n * Copyright
2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\nimport
getKClass\nimport kotlin.reflect.KClass\nimport kotlin.reflect.js.internal.KClassImpl\n\n/**\n * Represents the
constructor of a class. Instances of `JsClass` can be passed to JavaScript APIs that expect a constructor reference.\n
*/\n\nexternal interface JsClass<T : Any> {\n  /**\n   * Returns the unqualified name of the class represented by
this instance.\n   */\n   val name: String\n}\n\n/**\n * Obtains a constructor reference for the given `KClass`.\n
*/\n\nval <T : Any> KClass<T>.js: JsClass<T>\n  get() = (this as KClassImpl<T>).jClass\n\n/**\n * Obtains a
`KClass` instance for the given constructor reference.\n */\n\nval <T : Any> JsClass<T>.kotlin: KClass<T>\n  get()
= getKClass(this)\n", /*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.reflect.js.internal\n\nimport kotlin.reflect.*\n\ninternal abstract
class KClassImpl<T : Any>(\n  internal open val jClass: JsClass<T>\n) : KClass<T> {\n\n  override val
qualifiedName: String?\n  get() = TODO()\n\n  override fun equals(other: Any?): Boolean {\n    return other
is KClassImpl<*> && jClass == other.jClass\n  }\n\n  // TODO: use FQN\n  override fun hashCode(): Int =
simpleName?.hashCode() ?: 0\n\n  override fun toString(): String {\n    // TODO: use FQN\n    return "\"class
$simpleName\"\n  }\n}\n\ninternal class SimpleKClassImpl<T : Any>(jClass: JsClass<T>) :
KClassImpl<T>(jClass) {\n  override val simpleName: String? =
jClass.asDynamic().$metadata$.simpleName.unsafeCast<String?>()\n\n  override fun isInstance(value: Any?):
Boolean {\n    return jsIsType(value, jClass)\n  }\n}\n\ninternal class PrimitiveKClassImpl<T : Any>(\n
jClass: JsClass<T>,\n  private val givenSimpleName: String,\n  private val isInstanceFunction: (Any?) ->
Boolean\n) : KClassImpl<T>(jClass) {\n  override fun equals(other: Any?): Boolean {\n    if (other is
PrimitiveKClassImpl<*>) return false\n    return super.equals(other) && givenSimpleName ==
other.givenSimpleName\n  }\n\n  override val simpleName: String? get() = givenSimpleName\n\n  override fun
isInstance(value: Any?): Boolean {\n    return isInstanceFunction(value)\n  }\n}\n\ninternal object
NothingKClassImpl : KClassImpl<Nothing>(js("Object")) {\n  override val simpleName: String =
\"Nothing\"\n\n  override fun isInstance(value: Any?): Boolean = false\n\n  override val jClass:
JsClass<Nothing>\n  get() = throw UnsupportedOperationException(\"There's no native JS class for Nothing
type\")\n\n  override fun equals(other: Any?): Boolean = other === this\n\n  override fun hashCode(): Int =
0\n}\n\ninternal class ErrorKClass : KClass<Nothing> {\n  override val simpleName: String? get() =
error(\"Unknown simpleName for ErrorKClass\")\n\n  override val qualifiedName: String? get() = error(\"Unknown
qualifiedName for ErrorKClass\")\n\n  override fun isInstance(value: Any?): Boolean = error(\"Can't check
isInstance on ErrorKClass\")\n\n  override fun equals(other: Any?): Boolean = other === this\n\n  override fun
hashCode(): Int = 0\n}", /*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.reflect\n\ninternal actual inline val
KClass<*>.qualifiedOrSimpleName: String?\n  get() = simpleName", /*\n * Copyright 2010-2018 JetBrains s.r.o.
and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n */\n\n// a package is omitted to get declarations directly under
the module\n\n// TODO: Remove once JsReflectionAPIChecker supports more reflection
types\n\n@file:Suppress(\"Unsupported\")\n\nimport kotlin.reflect.*\nimport
kotlin.reflect.js.internal.*\n\n@JsName(\"createKType\")\n\ninternal fun createKType(\n  classifier: KClassifier,\n
arguments: Array<KTypeProjection>,\n  isMarkedNullable: Boolean\n) =\n  KTypeImpl(classifier,
arguments.asList(), isMarkedNullable)\n\n@JsName(\"createDynamicKType\")\n\ninternal fun
createDynamicKType(): KType = DynamicKType\n\n@JsName(\"markKTypeNullable\")\n\ninternal fun
markKTypeNullable(kType: KType) = KTypeImpl(kType.classifier!!, kType.arguments,
true)\n\n@JsName(\"createKTypeParameter\")\n\ninternal fun createKTypeParameter(\n  name: String,\n
upperBounds: Array<KType>,\n  variance: String\n): KTypeParameter {\n  val kVariance = when (variance) {\n

```

```

    |"in"| -> KVariance.IN\n    |"out"| -> KVariance.OUT\n    else -> KVariance.INVARIANT\n    }\n\n    return
KTypeParameterImpl(name, upperBounds.asList(), kVariance,
false)\n}\n\n@JsName("getStarKTypeProjection")\n\ninternal fun getStarKTypeProjection(): KTypeProjection =\nKTypeProjection.STAR\n\n@JsName("createCovariantKTypeProjection")\n\ninternal fun
createCovariantKTypeProjection(type: KType): KTypeProjection =\nKTypeProjection.covariant(type)\n\n@JsName("createInvariantKTypeProjection")\n\ninternal fun
createInvariantKTypeProjection(type: KType): KTypeProjection =\nKTypeProjection.invariant(type)\n\n@JsName("createContravariantKTypeProjection")\n\ninternal fun
createContravariantKTypeProjection(type: KType): KTypeProjection =\nKTypeProjection.contravariant(type)\n\n"/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.reflect.js.internal\n\nimport kotlin.reflect.*\n\ninternal class
KTypeImpl(\n    override val classifier: KClassifier,\n    override val arguments: List<KTypeProjection>,\n    override val isMarkedNullable: Boolean\n) : KType {\n    override fun equals(other: Any?): Boolean =\n        other
is KTypeImpl &&\n            classifier == other.classifier && arguments == other.arguments &&
isMarkedNullable == other.isMarkedNullable\n\n    override fun hashCode(): Int =\n        (classifier.hashCode() * 31
+ arguments.hashCode()) * 31 + isMarkedNullable.hashCode()\n\n    override fun toString(): String {\n        val
kClass = (classifier as? KClass<*>)\n        val classifierName = when {\n            kClass == null ->
classifier.toString()\n            kClass.simpleName != null -> kClass.simpleName\n            else -> |(non-denotable
type)|\n        }\n\n        val args =\n            if (arguments.isEmpty()) |"|"\n            else arguments.joinToString("|", "|",
|"<|", |">|")\n        val nullable = if (isMarkedNullable) |"?|" else |""\n        return classifierName + args +
nullable\n    }\n}\n\ninternal object DynamicKType : KType {\n    override val classifier: KClassifier? = null\n    override
val arguments: List<KTypeProjection> = emptyList()\n    override val isMarkedNullable: Boolean = false\n    override
fun toString(): String = |"dynamic"|}\n\n"/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\npackage kotlin.reflect.js.internal\n\nimport kotlin.reflect.*\n\ninternal
data class KTypeParameterImpl(\n    override val name: String,\n    override val upperBounds: List<KType>,\n    override
val variance: KVariance,\n    override val isReified: Boolean\n) : KTypeParameter {\n    override fun
toString(): String = name\n}\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.reflect.js.internal\n\nimport
kotlin.js.JsClass\n\n@JsName("PrimitiveClasses")\n\ninternal object PrimitiveClasses {\n    @JsName("anyClass")\n    val
anyClass = PrimitiveKClassImpl(js("Object").unsafeCast<JsClass<Any>>(),
|"Any"|, { it is Any })\n\n    @JsName("numberClass")\n    val numberClass =
PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Number>>(),
|"Number"|, { it is Number })\n\n    @JsName("nothingClass")\n    val nothingClass = NothingKClassImpl\n\n    @JsName("booleanClass")\n    val
booleanClass = PrimitiveKClassImpl(js("Boolean").unsafeCast<JsClass<Boolean>>(),
|"Boolean"|, { it is Boolean
})\n\n    @JsName("byteClass")\n    val byteClass =
PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Byte>>(),
|"Byte"|, { it is Byte })\n\n    @JsName("shortClass")\n    val shortClass = PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Short>>(),
|"Short"|, { it is Short })\n\n    @JsName("intClass")\n    val intClass =
PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Int>>(),
|"Int"|, { it is Int })\n\n    @JsName("floatClass")\n    val floatClass = PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Float>>(),
|"Float"|, { it is Float })\n\n    @JsName("doubleClass")\n    val doubleClass =
PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Double>>(),
|"Double"|, { it is Double })\n\n    @JsName("arrayClass")\n    val arrayClass =
PrimitiveKClassImpl(js("Array").unsafeCast<JsClass<Array<*>>(),
|"Array"|, { it is Array<*> })\n\n    @JsName("stringClass")\n    val stringClass = PrimitiveKClassImpl(js("String").unsafeCast<JsClass<String>>(),

```



```

/**\n * General category \"Cc\" in the Unicode specification.\n */\n CONTROL(15, \"Cc\"),\n\n /**\n *
General category \"Cf\" in the Unicode specification.\n */\n FORMAT(16, \"Cf\"),\n\n /**\n * General
category \"Co\" in the Unicode specification.\n */\n PRIVATE_USE(18, \"Co\"),\n\n /**\n * General
category \"Cs\" in the Unicode specification.\n */\n SURROGATE(19, \"Cs\"),\n\n /**\n * General category
\"Pd\" in the Unicode specification.\n */\n DASH_PUNCTUATION(20, \"Pd\"),\n\n /**\n * General
category \"Ps\" in the Unicode specification.\n */\n START_PUNCTUATION(21, \"Ps\"),\n\n /**\n *
General category \"Pe\" in the Unicode specification.\n */\n END_PUNCTUATION(22, \"Pe\"),\n\n /**\n *
General category \"Pc\" in the Unicode specification.\n */\n CONNECTOR_PUNCTUATION(23, \"Pc\"),\n\n
/**\n * General category \"Po\" in the Unicode specification.\n */\n OTHER_PUNCTUATION(24,
\"Po\"),\n\n /**\n * General category \"Sm\" in the Unicode specification.\n */\n MATH_SYMBOL(25,
\"Sm\"),\n\n /**\n * General category \"Sc\" in the Unicode specification.\n */\n
CURRENCY_SYMBOL(26, \"Sc\"),\n\n /**\n * General category \"Sk\" in the Unicode specification.\n */\n
MODIFIER_SYMBOL(27, \"Sk\"),\n\n /**\n * General category \"So\" in the Unicode specification.\n */\n
OTHER_SYMBOL(28, \"So\"),\n\n /**\n * General category \"Pi\" in the Unicode specification.\n */\n
INITIAL_QUOTE_PUNCTUATION(29, \"Pi\"),\n\n /**\n * General category \"Pf\" in the Unicode
specification.\n */\n FINAL_QUOTE_PUNCTUATION(30, \"Pf\");\n\n /**\n * Returns `true` if [char]
character belongs to this category.\n */\n public actual operator fun contains(char: Char): Boolean =
char.getCategoryValue() == this.value\n\n companion object {\n internal fun valueOf(category: Int):
CharCategory =\n when (category) {\n in 0..16 -> values()[category]\n in 18..30 ->
values()[category - 1]\n else -> throw IllegalArgumentException(\"Category #\$category is not defined.\")\n
}\n }\n\n\", /*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n
*/\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n */\n\npackage kotlin.text\n\n/**\n * The exception thrown when a character encoding or decoding error occurs.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual open class
CharacterCodingException(message: String?) : Exception(message) {\n actual constructor() : this(null)\n\n\", /*\n
*/\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.text\n\n/**\n * A mutable sequence of characters.\n */\n * String builder can be used to efficiently perform
multiple string manipulation operations.\n */\n\npublic actual class StringBuilder actual constructor(content: String) :
Appendable, CharSequence {\n /**\n * Constructs an empty string builder with the specified initial [capacity].\n
*/\n * In Kotlin/JS implementation of StringBuilder the initial capacity has no effect on the further performance
of operations.\n */\n actual constructor(capacity: Int) : this() {\n }\n\n /**\n * Constructs a string builder that
contains the same characters as the specified [content] char sequence.\n */\n actual constructor(content:
CharSequence) : this(content.toString()) {\n }\n\n /**\n * Constructs an empty string builder.\n */\n actual constructor() :
this(\"\")\n\n private var string: String = if (content != undefined) content else \"\"\n\n actual override val
length: Int\n get() = string.asDynamic().length\n\n actual override fun get(index: Int): Char =\n
string.getOrElse(index) { throw IndexOutOfBoundsException(\"index: \$index, length: \$length\") }\n\n actual
override fun subSequence(startIndex: Int, endIndex: Int): CharSequence = string.substring(startIndex, endIndex)\n\n
actual override fun append(value: Char): StringBuilder {\n string += value\n return this\n }\n\n actual
override fun append(value: CharSequence?): StringBuilder {\n string += value.toString()\n return this\n
}\n\n actual override fun append(value: CharSequence?, startIndex: Int, endIndex: Int): StringBuilder =\n
this.appendRange(value?: \"null\", startIndex, endIndex)\n\n /**\n * Reverses the contents of this string builder
and returns this instance.\n */\n * Surrogate pairs included in this string builder are treated as single
characters.\n * Therefore, the order of the high-low surrogates is never reversed.\n */\n * Note that the reverse
operation may produce new surrogate pairs that were unpaired low-surrogates and high-surrogates before the
operation.\n * For example, reversing `\"\\uDC00\\uD800\"` produces `\"\\uD800\\uDC00\"` which is a valid
surrogate pair.\n */\n actual fun reverse(): StringBuilder {\n var reversed = \"\"\n var index =
string.length - 1\n while (index >= 0) {\n val low = string[index--]\n if (low.isLowSurrogate()) &&

```

```

index >= 0) {\n          val high = string[index--]\n          if (high.isHighSurrogate()) {\n              reversed =
reversed + high + low\n          } else {\n              reversed = reversed + low + high\n          }\n      } else {\n          reversed += low\n      }\n      string = reversed\n      return this\n  }\n\n  /**\n   * Appends the string representation of the specified object [value] to this string builder and returns this instance.\n   *\n   * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n   * and then that string was appended to this string builder.\n   *\n   * actual fun append(value: Any?): StringBuilder\n  {\n      string += value.toString()\n      return this\n  }\n\n  /**\n   * Appends the string representation of the specified boolean [value] to this string builder and returns this instance.\n   *\n   * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n   * and then that string was appended to this string builder.\n   *\n   * actual fun append(value: Boolean): StringBuilder {\n      string += value\n      return this\n  }\n\n  /**\n   * Appends characters in the specified character array [value] to this string builder and returns this instance.\n   *\n   * Characters are appended in order, starting at the index 0.\n   *\n   * actual fun append(value: CharArray): StringBuilder {\n      string += value.concatToString()\n      return this\n  }\n\n  @Deprecated("Provided for binary compatibility.", level = DeprecationLevel.HIDDEN)\n  fun append(value: String): StringBuilder = append(value)\n\n  /**\n   * Appends the specified string [value] to this string builder and returns this instance.\n   *\n   * If [value] is `null`, then the four characters `\"null\"` are appended.\n   *\n   * actual fun append(value: String?): StringBuilder {\n      this.string += value ?: \"null\"\n      return this\n  }\n\n  /**\n   * Returns the current capacity of this string builder.\n   *\n   * The capacity is the maximum length this string builder can have before an allocation occurs.\n   *\n   * In Kotlin/JS implementation of StringBuilder the value returned from this method may not indicate the actual size of the backing storage.\n   *\n   * actual fun capacity(): Int = length\n\n  /**\n   * Ensures that the capacity of this string builder is at least equal to the specified [minimumCapacity].\n   *\n   * If the current capacity is less than the [minimumCapacity], a new backing storage is allocated with greater capacity.\n   *\n   * Otherwise, this method takes no action and simply returns.\n   *\n   * In Kotlin/JS implementation of StringBuilder the size of the backing storage is not extended to comply the given [minimumCapacity],\n   * thus calling this method has no effect on the further performance of operations.\n   *\n   * actual fun ensureCapacity(minimumCapacity: Int) {\n  }\n\n  /**\n   * Returns the index within this string builder of the first occurrence of the specified [string].\n   *\n   * Returns -1 if the specified [string] does not occur in this string builder.\n   *\n   * actual fun indexOf(string: String): Int = this.string.asDynamic().indexOf(string)\n\n  /**\n   * Returns the index within this string builder of the first occurrence of the specified [string],\n   * starting at the specified [startIndex].\n   *\n   * Returns -1 if the specified [string] does not occur in this string builder starting at the specified [startIndex].\n   *\n   * actual fun indexOf(string: String, startIndex: Int): Int = this.string.asDynamic().indexOf(string, startIndex)\n\n  /**\n   * Returns the index within this string builder of the last occurrence of the specified [string].\n   * The last occurrence of empty string `\"\"` is considered to be at the index equal to `this.length`.\n   *\n   * Returns -1 if the specified [string] does not occur in this string builder.\n   *\n   * actual fun lastIndexOf(string: String): Int = this.string.asDynamic().lastIndexOf(string)\n\n  /**\n   * Returns the index within this string builder of the last occurrence of the specified [string],\n   * starting from the specified [startIndex] toward the beginning.\n   *\n   * Returns -1 if the specified [string] does not occur in this string builder starting at the specified [startIndex].\n   *\n   * actual fun lastIndexOf(string: String, startIndex: Int): Int {\n      if (string.isEmpty() && startIndex < 0) return -1\n      return this.string.asDynamic().lastIndexOf(string, startIndex)\n  }\n\n  /**\n   * Inserts the string representation of the specified boolean [value] into this string builder at the specified [index] and returns this instance.\n   *\n   * The

```

```

overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n * and then
that string was inserted into this string builder at the specified [index].\n *\n * @throws
IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n
@SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun insert(index: Int, value:
Boolean): String Builder {\n     AbstractList.checkPositionIndex(index, length)\n\n     string = string.substring(0,
index) + value + string.substring(index)\n     return this\n }\n\n /**\n * Inserts the specified character [value]
into this string builder at the specified [index] and returns this instance.\n *\n * @throws
IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n
@SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun insert(index: Int, value:
Char): String Builder {\n     AbstractList.checkPositionIndex(index, length)\n\n     string = string.substring(0,
index) + value + string.substring(index)\n     return this\n }\n\n /**\n * Inserts characters in the specified
character array [value] into this string builder at the specified [index] and returns this instance.\n *\n * The
inserted characters go in same order as in the [value] character array, starting at [index].\n *\n * @throws
IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n
@SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun insert(index: Int, value:
CharArray): String Builder {\n     AbstractList.checkPositionIndex(index, length)\n\n     string =
string.substring(0, index) + value.concatToString() + string.substring(index)\n     return this\n }\n\n /**\n *
Inserts characters in the specified character sequence [value] into this string builder at the specified [index] and
returns this instance.\n *\n * The inserted characters go in the same order as in the [value] character sequence,
starting at [index].\n *\n * @param index the position in this string builder to insert at.\n * @param value the
character sequence from which characters are inserted. If [value] is `null`, then the four characters `"\u0000\u0000\u0000\u0000"` are
inserted.\n *\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of
this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun
insert(index: Int, value: CharSequence?): String Builder {\n     AbstractList.checkPositionIndex(index,
length)\n\n     string = string.substring(0, index) + value.toString() + string.substring(index)\n     return this\n
}\n\n /**\n * Inserts the string representation of the specified object [value] into this string builder at the
specified [index] and returns this instance.\n *\n * The overall effect is exactly as if the [value] were converted
to a string by the `value.toString()` method,\n * and then that string was inserted into this string builder at the
specified [index].\n *\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the
length of this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun
insert(index: Int, value: Any?): String Builder {\n     AbstractList.checkPositionIndex(index, length)\n\n     string = string.substring(0, index) + value.toString() +
string.substring(index)\n     return this\n }\n\n @Deprecated("Provided for binary compatibility.", level =
DeprecationLevel.HIDDEN)\n fun insert(index: Int, value: String): String Builder = insert(index, value)\n\n /**\n *
Inserts the string [value] into this string builder at the specified [index] and returns this instance.\n *\n * If
[value] is `null`, then the four characters `"\u0000\u0000\u0000\u0000"` are inserted.\n *\n * @throws IndexOutOfBoundsException
if [index] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n actual fun insert(index: Int, value: String?): String Builder
{\n     AbstractList.checkPositionIndex(index, length)\n\n     val toInsert = value ?: "\u0000\u0000\u0000\u0000"\n     this.string =
this.string.substring(0, index) + toInsert + this.string.substring(index)\n     return this\n }\n\n /**\n * Sets
the length of this string builder to the specified [newLength].\n *\n * If the [newLength] is less than the current
length, it is changed to the specified [newLength].\n * Otherwise, null characters `'\u0000'` are appended to this
string builder until its length is less than the [newLength].\n *\n * Note that in Kotlin/JS [set] operator function
has non-constant execution time complexity.\n * Therefore, increasing length of this string builder and then
updating each character by index may slow down your program.\n *\n * @throws
IndexOutOfBoundsException or [IllegalArgumentException] if [newLength] is less than zero.\n */\n
@SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun setLength(newLength:
Int) {\n     if (newLength < 0) {\n         throw IllegalArgumentException("Negative new length:

```



```

newLength.`\n    }\n    if (newLength <= length) {\n        string = string.substring(0, newLength)\n    }
else {\n    for (i in length until newLength) {\n        string += "\u0000"\n    }\n }\n }/**\n
 * Returns a new [String] that contains characters in this string builder at [startIndex] (inclusive) and up to the
[length] (exclusive).\n * \n * @throws IndexOutOfBoundsException if [startIndex] is less than zero or greater
than the length of this string builder.\n * \n * @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n actual fun substring(startIndex: Int): String {\n
AbstractList.checkPositionIndex(startIndex, length)\n    return string.substring(startIndex)\n }\n }/**\n
 * Returns a new [String] that contains characters in this string builder at [startIndex] (inclusive) and up to the
[endIndex] (exclusive).\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when
[startIndex] or [endIndex] is out of range of this string builder indices or when `startIndex > endIndex`.\n * \n
@SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun substring(startIndex:
Int, endIndex: Int): String {\n    AbstractList.checkBoundsIndexes(startIndex, endIndex, length)\n    return
string.substring(startIndex, endIndex)\n }\n }/**\n
 * Attempts to reduce storage used for this string builder.\n
 * \n * If the backing storage of this string builder is larger than necessary to hold its current contents,\n
 * then it may be resized to become more space efficient.\n
 * \n * Calling this method may, but is not required to, affect the value of the [capacity] property.\n
 * \n * In Kotlin/JS implementation of StringBuilder the size of the backing storage is always equal to the length of the string builder.\n
 * \n * @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n actual fun trimToSize() {\n }\n } override fun toString():
String = string\n }/**\n
 * Clears the content of this string builder making it empty and returns this instance.\n
 * \n * @sample samples.text.Strings.clearStringBuilder\n * \n * @SinceKotlin("1.3")\n public fun clear():
StringBuilder {\n    string = ""\n    return this\n }\n }/**\n
 * Sets the character at the specified [index] to the specified [value].\n
 * \n * @throws IndexOutOfBoundsException if [index] is out of bounds of this string builder.\n
 * \n * @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public operator fun set(index: Int, value: Char) {\n
AbstractList.checkElementIndex(index, length)\n    string = string.substring(0, index) + value + string.substring(index + 1)\n }\n }/**\n
 * Replaces characters in the specified range of this string builder with characters in the specified string [value] and returns this instance.\n
 * \n * @param startIndex the beginning (inclusive) of the range to replace.\n
 * \n * @param endIndex the end (exclusive) of the range to replace.\n
 * \n * @param value the string to replace with.\n
 * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] if [startIndex] is less than zero, greater than the length of this string builder, or `startIndex > endIndex`.\n
 * \n * @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n public fun setRange(startIndex: Int, endIndex: Int, value: String): StringBuilder {\n
checkReplaceRange(startIndex, endIndex, length)\n    this.string = this.string.substring(0, startIndex) + value + this.string.substring(endIndex)\n    return this\n }\n } private fun checkReplaceRange(startIndex: Int, endIndex: Int, length: Int) {\n
if (startIndex < 0 || startIndex > length) {\n    throw IndexOutOfBoundsException("`startIndex: $startIndex, length: $length`")\n }\n if (startIndex > endIndex) {\n
throw IllegalArgumentException("`startIndex($startIndex) > endIndex($endIndex)`")\n }\n }\n }/**\n
 * Removes the character at the specified [index] from this string builder and returns this instance.\n
 * \n * If the `Char` at the specified [index] is part of a supplementary code point, this method does not remove the entire supplementary character.\n
 * \n * @param index the index of `Char` to remove.\n
 * \n * @throws IndexOutOfBoundsException if [index] is out of bounds of this string builder.\n
 * \n * @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n public fun deleteAt(index: Int): StringBuilder {\n
AbstractList.checkElementIndex(index, length)\n    string = string.substring(0, index) + string.substring(index + 1)\n    return this\n }\n }/**\n
 * Removes characters in the specified range from this string builder and returns this instance.\n
 * \n * @param startIndex the beginning (inclusive) of the range to remove.\n
 * \n * @param endIndex the end (exclusive) of the range to remove.\n
 * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] is out of range of this string builder indices or when `startIndex > endIndex`.\n
 * \n * @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public fun deleteRange(startIndex: Int, endIndex: Int): StringBuilder {\n
checkReplaceRange(startIndex, endIndex,

```

```

length)\n\n    string = string.substring(0, startIndex) + string.substring(endIndex)\n    return this\n }\n\n /**\n  * Copies characters from this string builder into the [destination] character array.\n  *\n  * @param
destination the array to copy to.\n  * @param destinationOffset the position in the array to copy to, 0 by default.\n
  * @param startIndex the beginning (inclusive) of the range to copy, 0 by default.\n  * @param endIndex the end
(exclusive) of the range to copy, length of this string builder by default.\n  *\n  * @throws
IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this
string builder indices or when `startIndex > endIndex`.\n  * @throws IndexOutOfBoundsException when the
subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n  * or when that index
is out of the [destination] array indices range.\n  */\n @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n public fun toCharArray(destination: CharArray,
destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = this.length) {\n
AbstractList.checkBoundsIndexes(startIndex, endIndex, length)\n
AbstractList.checkBoundsIndexes(destinationOffset, destinationOffset + endIndex - startIndex, destination.size)\n\n
    var dstIndex = destinationOffset\n    for (index in startIndex until endIndex) {\n        destination[dstIndex++]
= string[index]\n    }\n }\n\n /**\n  * Appends characters in a subarray of the specified character array
[value] to this string builder and returns this instance.\n  *\n  * Characters are appended in order, starting at
specified [startIndex].\n  *\n  * @param value the array from which characters are appended.\n  * @param
startIndex the beginning (inclusive) of the subarray to append.\n  * @param endIndex the end (exclusive) of the
subarray to append.\n  *\n  * @throws IndexOutOfBoundsException or [IllegalArgumentException] when
[startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex > endIndex`.\n  */\n
@SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public fun appendRange(value:
CharArray, startIndex: Int, endIndex: Int): StringBuilder {\n    string += value.concatToString(startIndex,
endIndex)\n    return this\n }\n\n /**\n  * Appends a subsequence of the specified character sequence [value]
to this string builder and returns this instance.\n  *\n  * @param value the character sequence from which a
subsequence is appended.\n  * @param startIndex the beginning (inclusive) of the subsequence to append.\n  *
@param endIndex the end (exclusive) of the subsequence to append.\n  *\n  * @throws
IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the
[value] character sequence indices or when `startIndex > endIndex`.\n  */\n @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n public fun appendRange(value: CharSequence, startIndex:
Int, endIndex: Int): StringBuilder {\n    val stringCsq = value.toString()\n
AbstractList.checkBoundsIndexes(startIndex, endIndex, stringCsq.length)\n\n    string +=
stringCsq.substring(startIndex, endIndex)\n    return this\n }\n\n /**\n  * Inserts characters in a subarray of
the specified character array [value] into this string builder at the specified [index] and returns this instance.\n
*\n  * The inserted characters go in same order as in the [value] array, starting at [index].\n  *\n  * @param index
the position in this string builder to insert at.\n  * @param value the array from which characters are inserted.\n
*\n  * @param startIndex the beginning (inclusive) of the subarray to insert.\n  * @param endIndex the end (exclusive)
of the subarray to insert.\n  *\n  * @throws IndexOutOfBoundsException or [IllegalArgumentException] when
[startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex > endIndex`.\n  *
@throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n
*/\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public fun
insertRange(index: Int, value: CharArray, startIndex: Int, endIndex: Int): StringBuilder {\n
AbstractList.checkPositionIndex(index, this.length)\n\n    string = string.substring(0, index) +
value.concatToString(startIndex, endIndex) + string.substring(index)\n    return this\n }\n\n /**\n  * Inserts
characters in a subsequence of the specified character sequence [value] into this string builder at the specified
[index] and returns this instance.\n  *\n  * The inserted characters go in the same order as in the [value] character
sequence, starting at [index].\n  *\n  * @param index the position in this string builder to insert at.\n  *
@param value the character sequence from which a subsequence is inserted.\n  * @param startIndex the beginning
(inclusive) of the subsequence to insert.\n  * @param endIndex the end (exclusive) of the subsequence to insert.\n

```

```

    * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is
    out of range of the [value] character sequence indices or when `startIndex > endIndex`.
    * @throws
    IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
    *
    @SinceKotlin("1.4")
    @WasExperimental(ExperimentalStdlibApi::class)
    public fun insertRange(index: Int,
    value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder {
        AbstractList.checkPositionIndex(index,
        length)
        val stringCsq = value.toString()
        AbstractList.checkBoundsIndexes(startIndex, endIndex,
        stringCsq.length)
        string = string.substring(0, index) + stringCsq.substring(startIndex, endIndex) +
        string.substring(index)
        return this
    }
    * Clears the content of this string builder making it
    empty and returns this instance.
    *
    @sample samples.text.Strings.clearStringBuilder
    *
    @SinceKotlin("1.3")
    @Suppress("EXTENSION_SHADOWED_BY_MEMBER",
    "NOTHING_TO_INLINE")
    public actual inline fun StringBuilder.clear(): StringBuilder = this.clear()
    *
    * Sets the character at the specified [index] to the specified [value].
    * @throws IndexOutOfBoundsException if
    [index] is out of bounds of this string builder.
    *
    @SinceKotlin("1.4")
    @WasExperimental(ExperimentalStdlibApi::class)
    @Suppress("EXTENSION_SHA
    DOWED_BY_MEMBER", "NOTHING_TO_INLINE")
    public actual inline operator fun
    StringBuilder.set(index: Int, value: Char) = this.set(index, value)
    *
    * Replaces characters in the specified
    range of this string builder with characters in the specified string [value] and returns this instance.
    * @param
    startIndex the beginning (inclusive) of the range to replace.
    * @param endIndex the end (exclusive) of the range to
    replace.
    * @param value the string to replace with.
    * @throws IndexOutOfBoundsException or
    [IllegalArgumentException] if [startIndex] is less than zero, greater than the length of this string builder, or
    `startIndex > endIndex`.
    *
    @SinceKotlin("1.4")
    @WasExperimental(ExperimentalStdlibApi::class)
    @Suppress("EXTENSION_SHA
    DOWED_BY_MEMBER", "NOTHING_TO_INLINE")
    public actual inline fun
    StringBuilder.setRange(startIndex: Int, endIndex: Int, value: String): StringBuilder =
        this.setRange(startIndex,
        endIndex, value)
    *
    * Removes the character at the specified [index] from this string builder and returns this
    instance.
    *
    * If the `Char` at the specified [index] is part of a supplementary code point, this method does not
    remove the entire supplementary character.
    *
    * @param index the index of `Char` to remove.
    * @throws
    IndexOutOfBoundsException if [index] is out of bounds of this string builder.
    *
    @SinceKotlin("1.4")
    @WasExperimental(ExperimentalStdlibApi::class)
    @Suppress("EXTENSION_SHA
    DOWED_BY_MEMBER", "NOTHING_TO_INLINE")
    public actual inline fun
    StringBuilder.deleteAt(index:
    Int): StringBuilder = this.deleteAt(index)
    *
    * Removes characters in the specified range from this string
    builder and returns this instance.
    *
    * @param startIndex the beginning (inclusive) of the range to remove.
    *
    * @param endIndex the end (exclusive) of the range to remove.
    *
    * @throws IndexOutOfBoundsException or
    [IllegalArgumentException] when [startIndex] is out of range of this string builder indices or when `startIndex >
    endIndex`.
    *
    @SinceKotlin("1.4")
    @WasExperimental(ExperimentalStdlibApi::class)
    @Suppress("EXTENSION_SHA
    DOWED_BY_MEMBER", "NOTHING_TO_INLINE")
    public actual inline fun
    StringBuilder.deleteRange(startIndex: Int, endIndex: Int):
    StringBuilder = this.deleteRange(startIndex,
    endIndex)
    *
    * Copies characters from this string builder into the [destination] character array.
    *
    * @param destination the array to copy to.
    *
    * @param destinationOffset the position in the array to copy to, 0 by
    default.
    *
    * @param startIndex the beginning (inclusive) of the range to copy, 0 by default.
    *
    * @param endIndex
    the end (exclusive) of the range to copy, length of this string builder by default.
    *
    * @throws
    IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this
    string builder indices or when `startIndex > endIndex`.
    *
    * @throws IndexOutOfBoundsException when the
    subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
    *
    * or when that index is
    out of the [destination] array indices range.
    *
    @SinceKotlin("1.4")
    @WasExperimental(ExperimentalStdlibApi::class)
    @Suppress("EXTENSION_SHA
    DOWED_BY_MEMBER", "NOTHING_TO_INLINE",

```

```

\ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\)\npublic actual inline fun
StringBuilder.toCharArray(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int =
this.length) =\n    this.toCharArray(destination, destinationOffset, startIndex, endIndex)\n\n/**\n * Appends
characters in a subarray of the specified character array [value] to this string builder and returns this instance.\n *\n *
Characters are appended in order, starting at specified [startIndex].\n *\n * @param value the array from which
characters are appended.\n *\n * @param startIndex the beginning (inclusive) of the subarray to append.\n *\n * @param
endIndex the end (exclusive) of the subarray to append.\n *\n * @throws IndexOutOfBoundsException or
[IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] array indices or when
`startIndex > endIndex`.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("EXTENSION_SHA
DOWED_BY_MEMBER", "NOTHING_TO_INLINE")\npublic actual inline fun
StringBuilder.appendRange(value: CharArray, startIndex: Int, endIndex: Int): StringBuilder =\n
this.appendRange(value, startIndex, endIndex)\n\n/**\n * Appends a subsequence of the specified character
sequence [value] to this string builder and returns this instance.\n *\n * @param value the character sequence from
which a subsequence is appended.\n *\n * @param startIndex the beginning (inclusive) of the subsequence to append.\n
*\n * @param endIndex the end (exclusive) of the subsequence to append.\n *\n * @throws
IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the
[value] character sequence indices or when `startIndex > endIndex`.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("EXTENSION_SHA
DOWED_BY_MEMBER", "NOTHING_TO_INLINE")\npublic actual inline fun
StringBuilder.appendRange(value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder =\n
this.appendRange(value, startIndex, endIndex)\n\n/**\n * Inserts characters in a subarray of the specified character
array [value] into this string builder at the specified [index] and returns this instance.\n *\n * The inserted characters
go in same order as in the [value] array, starting at [index].\n *\n * @param index the position in this string builder
to insert at.\n *\n * @param value the array from which characters are inserted.\n *\n * @param startIndex the beginning
(inclusive) of the subarray to insert.\n *\n * @param endIndex the end (exclusive) of the subarray to insert.\n *\n *
@throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of
range of the [value] array indices or when `startIndex > endIndex`.\n *\n * @throws IndexOutOfBoundsException if
[index] is less than zero or greater than the length of this string builder.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("EXTENSION_SHA
DOWED_BY_MEMBER", "NOTHING_TO_INLINE")\npublic actual inline fun
StringBuilder.insertRange(index: Int, value: CharArray, startIndex: Int, endIndex: Int): StringBuilder =\n
this.insertRange(index, value, startIndex, endIndex)\n\n/**\n * Inserts characters in a subsequence of the specified
character sequence [value] into this string builder at the specified [index] and returns this instance.\n *\n * The
inserted characters go in the same order as in the [value] character sequence, starting at [index].\n *\n * @param
index the position in this string builder to insert at.\n *\n * @param value the character sequence from which a
subsequence is inserted.\n *\n * @param startIndex the beginning (inclusive) of the subsequence to insert.\n *\n * @param
endIndex the end (exclusive) of the subsequence to insert.\n *\n * @throws IndexOutOfBoundsException or
[IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] character sequence
indices or when `startIndex > endIndex`.\n *\n * @throws IndexOutOfBoundsException if [index] is less than zero or
greater than the length of this string builder.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("EXTENSION_SHA
DOWED_BY_MEMBER", "NOTHING_TO_INLINE")\npublic actual inline fun
StringBuilder.insertRange(index: Int, value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder =\n
this.insertRange(index, value, startIndex, endIndex)\n\n"/**\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n *\n@npackage kotlin.text\n\n/**\n * Returns `true` if the content of this
string is equal to the word `true`, ignoring case, and `false` otherwise.\n *\n@Deprecated("Use Kotlin compiler

```

```

1.4 to avoid deprecation warning.
@DeprecatedSinceKotlin(hiddenSince =
"1.4")
@kotlin.internal.InlineOnly
public actual inline fun String.toBoolean(): Boolean =
this.toBoolean()
Returns `true` if this string is not `null` and its content is equal to the word `true`,
ignoring case, and `false` otherwise.
There are also strict versions of the function available on non-nullable
String, [toBooleanStrict] and [toBooleanStrictOrNull].
@SinceKotlin("1.4")
public actual fun
String?.toBoolean(): Boolean = this != null && this.lowercase() == "true"
Parses the string as a signed
[Byte] number and returns the result.
@throws NumberFormatException if the string is not a valid
representation of a number.
public actual fun String.toByteArray(): Byte = toByteOrNull() ?:
numberFormatError(this)
Parses the string as a signed [Byte] number and returns the result.
@throws
NumberFormatException if the string is not a valid representation of a number.
@throws
IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
public actual fun
String.toByteArray(radix: Int): Byte = toByteOrNull(radix) ?: numberFormatError(this)
Parses the string as a
[Short] number and returns the result.
@throws NumberFormatException if the string is not a valid
representation of a number.
public actual fun String.toShort(): Short = toShortOrNull() ?:
numberFormatError(this)
Parses the string as a [Short] number and returns the result.
@throws
NumberFormatException if the string is not a valid representation of a number.
@throws
IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
public actual fun
String.toShort(radix: Int): Short = toShortOrNull(radix) ?: numberFormatError(this)
Parses the string as
an [Int] number and returns the result.
@throws NumberFormatException if the string is not a valid
representation of a number.
public actual fun String.toInt(): Int = toIntOrNull() ?:
numberFormatError(this)
Parses the string as an [Int] number and returns the result.
@throws
NumberFormatException if the string is not a valid representation of a number.
@throws
IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
public actual fun
String.toInt(radix: Int): Int = toIntOrNull(radix) ?: numberFormatError(this)
Parses the string as a [Long]
number and returns the result.
@throws NumberFormatException if the string is not a valid representation of a
number.
public actual fun String.toLong(): Long = toLongOrNull() ?: numberFormatError(this)
Parses the string as a [Long]
number and returns the result.
@throws NumberFormatException if the string is not
a valid representation of a number.
@throws IllegalArgumentException when [radix] is not a valid radix for
string to number conversion.
public actual fun
String.toLong(radix: Int): Long = toLongOrNull(radix) ?:
numberFormatError(this)
Parses the string as a [Double] number and returns the result.
@throws
NumberFormatException if the string is not a valid representation of a number.
public actual fun
String.toDouble(): Double = +(this.asDynamic()).unsafeCast<Double>().also {
    if (it.isNaN() && !this.isNaN()
|| it == 0.0 && this.isBlank())
        numberFormatError(this)
}
Parses the string as a [Float] number
and returns the result.
@throws NumberFormatException if the string is not a valid representation of a
number.
@kotlin.internal.InlineOnly
public actual inline fun String.toFloat(): Float =
toDouble().unsafeCast<Float>()
Parses the string as a [Double] number and returns the result
or `null`
if the string is not a valid representation of a number.
public actual fun String.toDoubleOrNull(): Double? =
+(this.asDynamic()).unsafeCast<Double>().takeIf {
    !(it.isNaN() && !this.isNaN() || it == 0.0 &&
this.isBlank())
}
Parses the string as a [Float] number and returns the result
or `null` if the string is
not a valid representation of a number.
@kotlin.internal.InlineOnly
public actual inline fun
String.toFloatOrNull(): Float? = toDoubleOrNull().unsafeCast<Float?>()
Returns a string representation
of this [Byte] value in the specified [radix].
@throws IllegalArgumentException when [radix] is not a valid
radix for number to string conversion.
@SinceKotlin("1.2")
@kotlin.internal.InlineOnly
public actual
inline fun Byte.toString(radix: Int): String = this.toInt().toString(radix)
Returns a string representation of
this [Short] value in the specified [radix].
@throws IllegalArgumentException when [radix] is not a valid
radix for number to string conversion.
@SinceKotlin("1.2")
@kotlin.internal.InlineOnly
public actual
inline fun Short.toString(radix: Int): String = this.toInt().toString(radix)
Returns a string representation of
this [Int] value in the specified [radix].
@throws IllegalArgumentException when [radix] is not a valid radix

```

```

for number to string conversion.\n *\n@SinceKotlin("1.2")\npublic actual fun Int.toString(radix: Int): String =
asDynamic().toString(checkRadix(radix))\n\nprivate fun String.isNaN(): Boolean = when (this.lowercase()) {\n
\n"nan", "+nan", "-nan" -> true\n else -> false\n}\n\n**\n * Checks whether the given [radix] is valid radix for
string to number and number to string conversion.\n *\n@PublishedApi\ninternal actual fun checkRadix(radix: Int):
Int {\n if (radix !in 2..36) {\n throw IllegalArgumentException("\radix $radix was not in valid range 2..36")\n
}\n return radix\n}\n\ninternal actual fun digitOf(char: Char, radix: Int): Int = when {\n char >= '0' && char <=
'9' -> char - '0'\n char >= 'A' && char <= 'Z' -> char - 'A' + 10\n char >= 'a' && char <= 'z' -> char - 'a' + 10\n
char < "\u0080" -> -1\n char >= "\uFF21" && char <= "\uFF3A" -> char - "\uFF21" + 10 // full-width latin capital
letter\n char >= "\uFF41" && char <= "\uFF5A" -> char - "\uFF41" + 10 // full-width latin small letter\n else ->
char.digitToIntImpl()\n}.let { if (it >= radix) -1 else it }\n", "\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n *\n\npackage kotlin.text\nimport kotlin.js.RegExp\n\n**\n * Provides
enumeration values to use to set regular expression options.\n *\npublic actual enum class RegexOptions(val value:
String) {\n /** Enables case-insensitive matching. *\n IGNORE_CASE("i"),\n /** Enables multiline
mode.\n *\n * In multiline mode the expressions `^` and `$` match just after or just before,\n * respectively, a
line terminator or the end of the input sequence. *\n MULTILINE("m")\n}\n\nprivate fun
Iterable<RegexOption>.toFlags(prepend: String): String = joinToString("\", prefix = prepend) { it.value
}\n\n**\n * Represents the results from a single capturing group within a [MatchResult] of [Regex].\n *\n *
@param value The value of captured group.\n *\npublic actual data class MatchGroup(actual val value:
String)\n\n**\n * Returns a named group with the specified [name].\n *\n * @return An instance of
[MatchGroup] if the group with the specified [name] was matched or `null` otherwise.\n * @throws
IllegalArgumentException if there is no group with the specified [name] defined in the regex pattern.\n * @throws
UnsupportedOperationException if this match group collection doesn't support getting match groups by name,\n *
for example, when it's not supported by the current platform.\n *\n@SinceKotlin("1.7")\npublic operator fun
MatchGroupCollection.get(name: String): MatchGroup? {\n val namedGroups = this as?
MatchNamedGroupCollection\n ?: throw UnsupportedOperationException("\Retrieving groups by name is not
supported on this platform.")\n return namedGroups[name]\n}\n\n**\n * Represents a compiled regular
expression.\n * Provides functions to match strings in text with a pattern, replace the found occurrences and split
text around matches.\n *\n * For pattern syntax reference see [MDN RegExp](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp#Special_characters_meaning_in_regular_expressions)\n
*\n * and
[http://www.w3schools.com/jsref/jsref_obj_regexp.asp](https://www.w3schools.com/jsref/jsref_obj_regexp.asp).\n
*\n * Note that `RegExp` objects under the hood are constructed with [the `u`
flag](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp/unicode)\n * that
enables Unicode-related features in regular expressions. This also makes the pattern syntax more strict,\n * for
example, prohibiting unnecessary escape sequences.\n *\n * @constructor Creates a regular expression from the
specified [pattern] string and the specified set of [options].\n *\npublic actual class Regex actual
constructor(pattern: String, options: Set<RegexOption>) {\n /** Creates a regular expression from the specified
[pattern] string and the specified single [option]. *\n public actual constructor(pattern: String, option:
RegexOption) : this(pattern, setOf(option))\n /** Creates a regular expression from the specified [pattern] string
and the default options. *\n public actual constructor(pattern: String) : this(pattern, emptySet())\n\n /** The
pattern string of this regular expression. *\n public actual val pattern: String = pattern\n /** The set of options
that were used to create this regular expression. *\n public actual val options: Set<RegexOption> =
options.toSet()\n private val nativePattern: RegExp = RegExp(pattern, options.toFlags("gu"))\n private var
nativeStickyPattern: RegExp? = null\n private fun initStickyPattern(): RegExp =\n nativeStickyPattern ?:
RegExp(pattern, options.toFlags("yu")).also { nativeStickyPattern = it }\n private var
nativeMatchesEntirePattern: RegExp? = null\n private fun initMatchesEntirePattern(): RegExp =\n
nativeMatchesEntirePattern ?: run {\n if (pattern.startsWith('^') && pattern.endsWith('$'))\n

```

```

nativePattern\n         else\n         return RegExp("\\^${pattern.trimStart('^').trimEnd('$')}\\$",
options.toFlags("\\gu"))\n     }.also { nativeMatchesEntirePattern = it }\n\n     /** Indicates whether the regular
expression matches the entire [input]. */\n     public actual infix fun matches(input: CharSequence): Boolean {\n
nativePattern.reset()\n     val match = nativePattern.exec(input.toString())\n     return match != null &&
match.index == 0 && nativePattern.lastIndex == input.length\n     }\n\n     /** Indicates whether the regular
expression can find at least one match in the specified [input]. */\n     public actual fun containsMatchIn(input:
CharSequence): Boolean {\n     nativePattern.reset()\n     return nativePattern.test(input.toString())\n     }\n\n
@SinceKotlin("\\1.7")\n     @WasExperimental(ExperimentalStdlibApi::class)\n     public actual fun
matchesAt(input: CharSequence, index: Int): Boolean {\n     if (index < 0 || index > input.length) {\n     throw
IndexOutOfBoundsException("index out of bounds: $index, input length: ${input.length}")\n     }\n     val
pattern = initStickyPattern()\n     pattern.lastIndex = index\n     return pattern.test(input.toString())\n     }\n\n
/**\n     * Returns the first match of a regular expression in the [input], beginning at the specified [startIndex].\n
*\n     * @param startIndex An index to start search with, by default 0. Must be not less than zero and not greater
than `input.length()`\n     * @return An instance of [MatchResult] if match was found or `null` otherwise.\n     *
@throws IndexOutOfBoundsException if [startIndex] is less than zero or greater than the length of the [input] char
sequence.\n     * @sample samples.text.Regexps.find\n     */\n
@Suppress("\\ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n     public actual fun find(input:
CharSequence, startIndex: Int = 0): MatchResult? {\n     if (startIndex < 0 || startIndex > input.length) {\n
throw IndexOutOfBoundsException("Start index out of bounds: $startIndex, input length: ${input.length}")\n
}\n     return nativePattern.findNext(input.toString(), startIndex, nativePattern)\n     }\n\n     /**\n     * Returns a
sequence of all occurrences of a regular expression within the [input] string, beginning at the specified
[startIndex].\n     *\n     * @throws IndexOutOfBoundsException if [startIndex] is less than zero or greater than the
length of the [input] char sequence.\n     *\n     * @sample samples.text.Regexps.findAll\n     */\n
@Suppress("\\ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n     public actual fun findAll(input:
CharSequence, startIndex: Int = 0): Sequence<MatchResult> {\n     if (startIndex < 0 || startIndex > input.length)
{\n     throw IndexOutOfBoundsException("Start index out of bounds: $startIndex, input length:
${input.length}")\n     }\n     return generateSequence({ find(input, startIndex) }, { match -> match.next() })\n
}\n\n     /**\n     * Attempts to match the entire [input] CharSequence against the pattern.\n     *\n     * @return An
instance of [MatchResult] if the entire input matches or `null` otherwise.\n     */\n     public actual fun
matchEntire(input: CharSequence): MatchResult? =\n     initMatchesEntirePattern().findNext(input.toString(), 0,
nativePattern)\n\n     @SinceKotlin("\\1.7")\n     @WasExperimental(ExperimentalStdlibApi::class)\n     public actual
fun matchAt(input: CharSequence, index: Int): MatchResult? {\n     if (index < 0 || index > input.length) {\n
throw IndexOutOfBoundsException("index out of bounds: $index, input length: ${input.length}")\n     }\n
return initStickyPattern().findNext(input.toString(), index, nativePattern)\n     }\n\n     /**\n     * Replaces all
occurrences of this regular expression in the specified [input] string with specified [replacement] expression.\n     *
The replacement string may contain references to the captured groups during a match. Occurrences of `${name}`
or `${index}`\n     * in the replacement string will be substituted with the subsequences corresponding to the captured
groups with the specified name or index.\n     * In case of `${index}`, the first digit after '$' is always treated as a part
of group reference. Subsequent digits are incorporated\n     * into `index` only if they would form a valid group
reference. Only the digits '0'..'9' are considered as potential components\n     * of the group reference. Note that
indexes of captured groups start from 1, and the group with index 0 is the whole match.\n     * In case of `${name}`,
the `name` can consist of latin letters 'a'..'z' and 'A'..'Z', or digits '0'..'9'. The first character must be\n     * a letter.\n
*\n     * Backslash character '\\' can be used to include the succeeding character as a literal in the replacement string,
e.g. '\\$' or '\\\\\\'.\n     * [Regex.escapeReplacement] can be used if [replacement] have to be treated as a literal
string.\n     *\n     * @param input the char sequence to find matches of this regular expression in\n     * @param
replacement the expression to replace found matches with\n     * @return the result of replacing each occurrence of
this regular expression in [input] with the result of evaluating the [replacement] expression\n     * @throws
RuntimeException if [replacement] expression is malformed, or capturing group with specified `name` or `index`

```

```

does not exist\n    */\n    public actual fun replace(input: CharSequence, replacement: String): String {\n        if
(!replacement.contains("\\\\") && !replacement.contains('$')) {\n            return
input.toString().nativeReplace(nativePattern, replacement)\n        }\n        return replace(input) {\n
substituteGroupRefs(it, replacement) }\n    }\n\n    /**\n     * Replaces all occurrences of this regular expression in
the specified [input] string with the result of\n     * the given function [transform] that takes [MatchResult] and
returns a string to be used as a\n     * replacement for that match.\n     */\n    public actual fun replace(input:
CharSequence, transform: (MatchResult) -> CharSequence): String {\n        var match = find(input)\n        if (match
== null) return input.toString()\n\n        var lastStart = 0\n        val length = input.length\n        val sb =
StringBuilder(length)\n        do {\n            val foundMatch = match!!\n            sb.append(input, lastStart,
foundMatch.range.start)\n            sb.append(transform(foundMatch))\n            lastStart =
foundMatch.range.endInclusive + 1\n            match = foundMatch.next()\n        } while (lastStart < length && match
!= null)\n\n        if (lastStart < length) {\n            sb.append(input, lastStart, length)\n        }\n\n        return
sb.toString()\n    }\n\n    /**\n     * Replaces the first occurrence of this regular expression in the specified [input]
string with specified [replacement] expression.\n     * The replacement string may contain references to the
captured groups during a match. Occurrences of `${name}` or `${index}`\n     * in the replacement string will be
substituted with the subsequences corresponding to the captured groups with the specified name or index.\n     * In
case of `${index}`, the first digit after '$' is always treated as a part of group reference. Subsequent digits are
incorporated\n     * into `index` only if they would form a valid group reference. Only the digits '0'..'9' are considered
as potential components\n     * of the group reference. Note that indexes of captured groups start from 1, and the
group with index 0 is the whole match.\n     * In case of `${name}`, the `name` can consist of latin letters 'a'..'z' and
'A'..'Z', or digits '0'..'9'. The first character must be\n     * a letter.\n     * Backslash character '\\' can be used to
include the succeeding character as a literal in the replacement string, e.g. `\\$` or `\\\\`.\n     *
[Regex.escapeReplacement] can be used if [replacement] have to be treated as a literal string.\n     * @param
input the char sequence to find a match of this regular expression in\n     * @param replacement the expression to
replace the found match with\n     * @return the result of replacing the first occurrence of this regular expression in
[input] with the result of evaluating the [replacement] expression\n     * @throws RuntimeException if
[replacement] expression is malformed, or capturing group with specified `name` or `index` does not exist\n     */\n
public actual fun replaceFirst(input: CharSequence, replacement: String): String {\n        if
(!replacement.contains("\\\\") && !replacement.contains('$')) {\n            val nonGlobalOptions =
options.toFlags("\\u")\n            return input.toString().nativeReplace(RegExp(pattern, nonGlobalOptions),
replacement)\n        }\n\n        val match = find(input) ?: return input.toString()\n        return buildString {\n
append(input.substring(0, match.range.first))\n            append(substituteGroupRefs(match, replacement))\n
append(input.substring(match.range.last + 1, input.length))\n        }\n    }\n\n    /**\n     * Splits the [input]
CharSequence to a list of strings around matches of this regular expression.\n     * @param limit Non-negative
value specifying the maximum number of substrings the string can be split to.\n     * Zero by default means no limit
is set.\n     */\n    @Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n    public actual fun
split(input: CharSequence, limit: Int = 0): List<String> {\n        requireNonNegativeLimit(limit)\n        val matches =
findAll(input).let { if (limit == 0) it else it.take(limit - 1) }\n        val result = mutableListOf<String>()\n        var
lastStart = 0\n        for (match in matches) {\n            result.add(input.subSequence(lastStart,
match.range.start).toString())\n            lastStart = match.range.endInclusive + 1\n        }\n        result.add(input.subSequence(lastStart, input.length).toString())\n        return result\n    }\n\n    /**\n     * Splits the
[input] CharSequence to a sequence of strings around matches of this regular expression.\n     * @param limit
Non-negative value specifying the maximum number of substrings the string can be split to.\n     * Zero by default
means no limit is set.\n     * @sample samples.text.Regexps.splitToSequence\n     */\n    @SinceKotlin("1.6")\n    @WasExperimental(ExperimentalStdlibApi::class)\n
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n    public actual fun
splitToSequence(input: CharSequence, limit: Int = 0): Sequence<String> {\n        requireNonNegativeLimit(limit)\n        return sequence {\n            var match = find(input)\n            if (match ==

```



```

substituteGroupRefs(match: MatchResult, replacement: String): String {
    var index = 0
    val result =
    StringBuilder()
    while (index < replacement.length) {
        val char = replacement[index++]
        if (char ==
        '\\') {
            if (index == replacement.length)
                throw IllegalArgumentException("The Char to be
                escaped is missing")
            result.append(replacement[index++])
        } else if (char == '$') {
            if
            (index == replacement.length)
                throw IllegalArgumentException("Capturing group index is
                missing")
            if (replacement[index] == '{') {
                val endIndex =
                replacement.readGroupName(++index)
                if (index == endIndex)
                    throw
                    IllegalArgumentException("Named capturing group reference should have a non-empty name")
                if
                (endIndex == replacement.length || replacement[endIndex] != '}')
                    throw
                    IllegalArgumentException("Named capturing group reference is missing trailing '}')
                val groupName
                = replacement.substring(index, endIndex)
                result.append(match.groups[groupName]?.value ?: "\\")
                index = endIndex + 1 // skip past '}'
            } else {
                if (replacement[index] !in '0'..'9')
                    throw IllegalArgumentException("Invalid capturing group reference")
                val groups = match.groups
                val endIndex = replacement.readGroupIndex(index, groups.size)
                val groupIndex =
                replacement.substring(index, endIndex).toInt()
                if (groupIndex >= groups.size)
                    throw
                    IndexOutOfBoundsException("Group with index $groupIndex does not exist")
                result.append(groups[groupIndex]?.value ?: "\\")
                index = endIndex
            }
        } else {
            result.append(char)
        }
    }
    return result.toString()
}
// The name must be a legal JavaScript identifier.
See https://262.ecma-international.org/5.1/#sec-7.6 Don't try to validate the referenced group name as it may be
time-consuming. If the name is invalid, it won't be found in `match.groups` anyway and will throw. Group
names in the target Regex are validated at creation time.
private fun String.readGroupName(startIndex: Int): Int {
    var index = startIndex
    while (index < length) {
        if (this[index] == '{')
            break
        } else {
            index++
        }
    }
    return index
}
private fun String.readGroupIndex(startIndex: Int, groupCount: Int):
Int {
    // at least one digit after '$' is always captured
    var index = startIndex + 1
    var groupIndex =
    this[startIndex] - '0'
    // capture the largest valid group index
    while (index < length && this[index] in '0'..'9') {
        val newGroupIndex = (groupIndex * 10) + (this[index] - '0')
        if (newGroupIndex in 0 until
        groupCount) {
            groupIndex = newGroupIndex
            index++
        } else {
            break
        }
    }
    return index
}
", "/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.
*/
@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("StringsKt")
@file:Suppress("EXTENSION_SHADOWED_BY_MEMBER")
package kotlin.text
import kotlin.contracts.*
 * A mutable
sequence of characters.
 * String builder can be used to efficiently perform multiple string manipulation
operations.
*/
expect class StringBuilder : Appendable, CharSequence {
    /** Constructs an empty string
builder.
 * constructor()
    /** Constructs an empty string builder with the specified initial [capacity].
 * constructor(capacity: Int)
    /** Constructs a string builder that contains the same characters as the specified
[content] char sequence.
 * constructor(content: CharSequence)
    /** Constructs a string builder that
contains the same characters as the specified [content] string.
 * @SinceKotlin("1.3")
 * @ExperimentalStdlibApi
    constructor(content: String)
    override val length: Int
    override operator fun
get(index: Int): Char
    override fun subSequence(startIndex: Int, endIndex: Int): CharSequence
    override
fun append(value: Char): StringBuilder
    override fun append(value: CharSequence?): StringBuilder
    override
fun append(value: CharSequence?, startIndex: Int, endIndex: Int): StringBuilder
    /**
 * Reverses the
contents of this string builder and returns this instance.
 * reverse()
    * Surrogate pairs included in this string builder
are treated as single characters.
 * Therefore, the order of the high-low surrogates is never reversed.
 * Note that the reverse operation may produce new surrogate pairs that were unpaired low-surrogates and high-
surrogates before the operation.
 * For example, reversing "\uDC00\uD800" produces "\uD800\uDC00"
which is a valid surrogate pair.
 * fun reverse(): StringBuilder
    /**
 * Appends the string
representation of the specified object [value] to this string builder and returns this instance.
 * append()
    * The overall

```

```

effect is exactly as if the [value] were converted to a string by the `value.toString()` method,
 * and then that string was appended to this string builder.
 * fun append(value: Any?): StringBuilder
 * Append the string representation of the specified boolean [value] to this string builder and returns this instance.
 * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,
 * and then that string was appended to this string builder.
 * fun append(value: Boolean): StringBuilder
 * Append characters in the specified character array [value] to this string builder and returns this instance.
 * Characters are appended in order, starting at the index 0.
 * fun append(value: CharArray): StringBuilder
 * Append the specified string [value] to this string builder and returns this instance.
 * If [value] is `null`, then the four characters `"\null"` are appended.
 * fun append(value: String?): StringBuilder
 * Returns the current capacity of this string builder.
 * The capacity is the maximum length this string builder can have before an allocation occurs.
 * fun capacity(): Int
 * Ensures that the capacity of this string builder is at least equal to the specified [minimumCapacity].
 * If the current capacity is less than the [minimumCapacity], a new backing storage is allocated with greater capacity.
 * Otherwise, this method takes no action and simply returns.
 * fun ensureCapacity(minimumCapacity: Int)
 * Returns the index within this string builder of the first occurrence of the specified [string].
 * Returns -1 if the specified [string] does not occur in this string builder.
 * fun indexOf(string: String): Int
 * Returns the index within this string builder of the first occurrence of the specified [string],
 * starting at the specified [startIndex].
 * Returns -1 if the specified [string] does not occur in this string builder starting at the specified [startIndex].
 * fun indexOf(string: String, startIndex: Int): Int
 * Returns the index within this string builder of the last occurrence of the specified [string].
 * The last occurrence of empty string `""` is considered to be at the index equal to `this.length`.
 * Returns -1 if the specified [string] does not occur in this string builder.
 * fun lastIndexOf(string: String): Int
 * Returns the index within this string builder of the last occurrence of the specified [string],
 * starting from the specified [startIndex] toward the beginning.
 * Returns -1 if the specified [string] does not occur in this string builder starting at the specified [startIndex].
 * fun lastIndexOf(string: String, startIndex: Int): Int
 * Inserts the string representation of the specified boolean [value] into this string builder at the specified [index] and returns this instance.
 * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,
 * and then that string was inserted into this string builder at the specified [index].
 * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
 * fun insert(index: Int, value: Boolean): StringBuilder
 * Inserts the specified character [value] into this string builder at the specified [index] and returns this instance.
 * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
 * fun insert(index: Int, value: Char): StringBuilder
 * Inserts characters in the specified character array [value] into this string builder at the specified [index] and returns this instance.
 * The inserted characters go in same order as in the [value] character array, starting at [index].
 * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
 * fun insert(index: Int, value: CharArray): StringBuilder
 * Inserts characters in the specified character sequence [value] into this string builder at the specified [index] and returns this instance.
 * The inserted characters go in the same order as in the [value] character sequence, starting at [index].
 * @param index

```

the position in this string builder to insert at.\n * @param value the character sequence from which characters are inserted. If [value] is `null`, then the four characters `\\null` are inserted.\n * @throws

IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun insert(index: Int, value: CharSequence?): StringBuilder\n\n /**\n * Inserts the string representation of the specified object [value] into this string builder at the specified [index] and returns this instance.\n * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n * and then that string was inserted into this string builder at the specified [index].\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun insert(index: Int, value: Any?): StringBuilder\n\n /**\n * Inserts the string [value] into this string builder at the specified [index] and returns this instance.\n * If [value] is `null`, then the four characters `\\null` are inserted.\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun insert(index: Int, value: String?): StringBuilder\n\n /**\n * Sets the length of this string builder to the specified [newLength].\n * If the [newLength] is less than the current length, it is changed to the specified [newLength].\n * Otherwise, null characters `\\u0000` are appended to this string builder until its length is less than the [newLength].\n * Note that in Kotlin/JS [set] operator function has non-constant execution time complexity.\n * Therefore, increasing length of this string builder and then updating each character by index may slow down your program.\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] if [newLength] is less than zero.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun setLength(newLength: Int)\n\n /**\n * Returns a new [String] that contains characters in this string builder at [startIndex] (inclusive) and up to the [length] (exclusive).\n * @throws IndexOutOfBoundsException if [startIndex] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun substring(startIndex: Int): String\n\n /**\n * Returns a new [String] that contains characters in this string builder at [startIndex] (inclusive) and up to the [endIndex] (exclusive).\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this string builder indices or when `startIndex > endIndex`.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun substring(startIndex: Int, endIndex: Int): String\n\n /**\n * Attempts to reduce storage used for this string builder.\n * If the backing storage of this string builder is larger than necessary to hold its current contents,\n * then it may be resized to become more space efficient.\n * Calling this method may, but is not required to, affect the value of the [capacity] property.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun trimToSize()\n\n /**\n * Clears the content of this string builder making it empty and returns this instance.\n */\n @sample samples.text.Strings.clearStringBuilder\n */\n @SinceKotlin("1.3")\n public expect fun StringBuilder.clear(): StringBuilder\n\n /**\n * Sets the character at the specified [index] to the specified [value].\n * @throws IndexOutOfBoundsException if [index] is out of bounds of this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public expect operator fun StringBuilder.set(index: Int, value: Char)\n\n /**\n * Replaces characters in the specified range of this string builder with characters in the specified string [value] and returns this instance.\n * @param startIndex the beginning (inclusive) of the range to replace.\n * @param endIndex the end (exclusive) of the range to replace.\n * @param value the string to replace with.\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] if [startIndex] is less than zero, greater than the length of this string builder, or `startIndex > endIndex`.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public expect fun StringBuilder.setRange(startIndex: Int, endIndex: Int, value: String): StringBuilder\n\n /**\n * Removes the character at the specified [index] from this string builder and returns this instance.\n * If the `Char` at the specified [index] is part of a supplementary code point, this method does not remove the entire supplementary character.\n * @param index the index of `Char` to remove.\n * @throws IndexOutOfBoundsException if

[index] is out of bounds of this string builder.\n

```

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
StringBuilder.deleteAt(index: Int): StringBuilder\n\n/**\n * Removes characters in the specified range from this
string builder and returns this instance.\n * \n * @param startIndex the beginning (inclusive) of the range to
remove.\n * @param endIndex the end (exclusive) of the range to remove.\n * \n * @throws
IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] is out of range of this string builder
indices or when `startIndex > endIndex`.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
StringBuilder.deleteRange(startIndex: Int, endIndex: Int): StringBuilder\n\n/**\n * Copies characters from this
string builder into the [destination] character array.\n * \n * @param destination the array to copy to.\n * @param
destinationOffset the position in the array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive)
of the range to copy, 0 by default.\n * @param endIndex the end (exclusive) of the range to copy, length of this
string builder by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when
[startIndex] or [endIndex] is out of range of this string builder indices or when `startIndex > endIndex`.\n * @throws
IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified
[destinationOffset],\n * or when that index is out of the [destination] array indices range.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
StringBuilder.toCharArray(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int =
this.length)\n\n/**\n * Appends characters in a subarray of the specified character array [value] to this string
builder
and returns this instance.\n * \n * Characters are appended in order, starting at specified [startIndex].\n * \n * @param
value the array from which characters are appended.\n * @param startIndex the beginning (inclusive) of the
subarray to append.\n * @param endIndex the end (exclusive) of the subarray to append.\n * \n * @throws
IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the
[value] array indices or when `startIndex > endIndex`.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
StringBuilder.appendRange(value: CharArray, startIndex: Int, endIndex: Int): StringBuilder\n\n/**\n * Appends a
subsequence of the specified character sequence [value] to this string builder and returns this instance.\n * \n *
@param value the character sequence from which a subsequence is appended.\n * @param startIndex the beginning
(inclusive) of the subsequence to append.\n * @param endIndex the end (exclusive) of the subsequence to append.\n
*\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out
of range of the [value] character sequence indices or when `startIndex > endIndex`.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
StringBuilder.appendRange(value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder\n\n/**\n * Inserts
characters in a subarray of the specified character array [value] into this string builder at the specified [index]
and
returns this instance.\n * \n * The inserted characters go in same order as in the [value] array, starting at
[index].\n * \n * @param index the position in this string builder to insert at.\n * @param value the array from
which characters
are inserted.\n * @param startIndex the beginning (inclusive) of the subarray to insert.\n * @param endIndex the
end (exclusive) of the subarray to insert.\n * \n * @throws IndexOutOfBoundsException or
[IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] array indices or when
`startIndex > endIndex`.\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the
length of this string builder.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
StringBuilder.insertRange(index: Int, value: CharArray, startIndex: Int, endIndex: Int): StringBuilder\n\n/**\n *
Inserts characters in a subsequence of the specified character sequence [value] into this string builder at the
specified
[index] and returns this instance.\n * \n * The inserted characters go in the same order as in the [value]
character
sequence, starting at [index].\n * \n * @param index the position in this string builder to insert at.\n * @param
value the character sequence from which a subsequence is inserted.\n * @param startIndex the beginning (inclusive)
of the
subsequence to insert.\n * @param endIndex the end (exclusive) of the subsequence to insert.\n * \n * @throws

```

```

IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the
[value] character sequence indices or when `startIndex > endIndex`.n * @throws IndexOutOfBoundsException if
[index] is less than zero or greater than the length of this string builder.n
*/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)npublic expect fun
StringBuilder.insertRange(index: Int, value: CharSequence, startIndex: Int, endIndex: Int):
StringBuilder\n\n@Suppress("EXTENSION_SHADOWED_BY_MEMBER")n@Deprecated("Use
append(value: Any?) instead", ReplaceWith("append(value = obj)"),
DeprecationLevel.WARNING)n@kotlin.internal.InlineOnlynpublic inline fun StringBuilder.append(obj: Any?):
StringBuilder = this.append(obj)\n\n/**n * Builds new string by populating newly created [StringBuilder] using
provided [builderAction]\n * and then converting it to [String].n */n@kotlin.internal.InlineOnlynpublic inline fun
buildString(builderAction: String Builder.() -> Unit): String {n contract { callsInPlace(builderAction,
InvocationKind.EXACTLY_ONCE) }n return StringBuilder().apply(builderAction).toString()\n}\n\n/**n *
Builds new string by populating newly created [StringBuilder] initialized with the given [capacity]\n * using
provided [builderAction] and then converting it to [String].n
*/n@SinceKotlin("1.1")n@kotlin.internal.InlineOnlynpublic inline fun buildString(capacity: Int, builderAction:
String Builder.() -> Unit): String {n contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE)
}\n return StringBuilder(capacity).apply(builderAction).toString()\n}\n\n/**n * Appends all arguments to the
given String Builder.n */npublic fun StringBuilder.append(vararg value: String?): String Builder {n for (item in
value)\n append(item)\n return this\n}\n\n/**n * Appends all arguments to the given String Builder.n
*/npublic fun StringBuilder.append(vararg value: Any?): String Builder {n for (item in value)\n
append(item)\n return this\n}\n\n/** Appends a line feed character (`\n`) to this String Builder.
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun StringBuilder.appendLine():
StringBuilder = append("\n")n\n/** Appends [value] to this [StringBuilder], followed by a line feed character
(`\n`). */n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun StringBuilder.appendLine(value:
CharSequence?): String Builder = append(value).appendLine()\n\n/** Appends [value] to this [StringBuilder],
followed by a line feed character (`\n`). */n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun
StringBuilder.appendLine(value: String?): String Builder = append(value).appendLine()\n\n/** Appends [value] to
this [StringBuilder], followed by a line feed character (`\n`).
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun StringBuilder.appendLine(value: Any?):
StringBuilder = append(value).appendLine()\n\n/** Appends [value] to this [StringBuilder], followed by a line feed
character (`\n`). */n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun
StringBuilder.appendLine(value: CharArray): String Builder = append(value).appendLine()\n\n/** Appends [value]
to this [StringBuilder], followed by a line feed character (`\n`).
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun StringBuilder.appendLine(value: Char):
StringBuilder = append(value).appendLine()\n\n/** Appends [value] to this [StringBuilder], followed by a line feed
character (`\n`). */n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun
StringBuilder.appendLine(value: Boolean): String Builder = append(value).appendLine()\n\n/*\n * Copyright 2010-
2021 JetBrains s.r.o. and Kotlin Programming Language contributors.n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.n */npackage kotlin.text\nimport
kotlin.js.RegExp\n\n@kotlin.internal.InlineOnlyninternal actual inline fun String.nativeIndexOf(ch: Char,
fromIndex: Int): Int = nativeIndexOf(ch.toString(), fromIndex)\n\n@kotlin.internal.InlineOnlyninternal actual
inline fun String.nativeLastIndexOf(ch: Char, fromIndex: Int): Int = nativeLastIndexOf(ch.toString(),
fromIndex)\n\n/**n * Returns `true` if this string starts with the specified prefix.n
*/n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")npublic actual fun
String.startsWith(prefix: String, ignoreCase: Boolean = false): Boolean {n if (!ignoreCase)\n return
nativeStartsWith(prefix, 0)\n else\n return regionMatches(0, prefix, 0, prefix.length, ignoreCase)\n}\n\n/**n *
Returns `true` if a substring of this string starting at the specified offset [startIndex] starts with the specified
prefix.n
*/n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")npublic actual fun

```

```

String.startsWith(prefix: String, startIndex: Int, ignoreCase: Boolean = false): Boolean {
    if (!ignoreCase)
        return nativeStartsWith(prefix, startIndex)
    else
        return regionMatches(startIndex, prefix, 0, prefix.length, ignoreCase)
}

Returns `true` if this string ends with the specified suffix.

@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.endsWith(suffix: String, ignoreCase: Boolean = false): Boolean {
    if (!ignoreCase)
        return nativeEndsWith(suffix)
    else
        return regionMatches(length - suffix.length, suffix, 0, suffix.length, ignoreCase)
}

@Deprecated("Use Regex.matches() instead", ReplaceWith("regex.toRegex().matches(this)"))
@DeprecatedSinceKotlin(warningSince = "1.6")
public fun String.matches(regex: String): Boolean {
    @Suppress("DEPRECATION")
    val result = this.match(regex)
    return result != null && result.size != 0
}

Returns `true` if this string is empty or consists solely of whitespace characters.

@sample samples.text.Strings.stringIsBlank
public actual fun CharSequence.isBlank(): Boolean = length == 0 || indices.all { this[it].isWhitespace() }

Returns `true` if this string is equal to [other], optionally ignoring character case.

Two strings are considered to be equal if they have the same length and the same character at the same index.

If [ignoreCase] is true, the result of `Char.toUpperCaseChar().toLowerCaseChar()` on each character is compared.

@param ignoreCase `true` to ignore character case when comparing strings. By default `false`.

@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String?.equals(other: String?, ignoreCase: Boolean = false): Boolean {
    if (this == null) return other == null
    if (other == null) return false
    if (!ignoreCase) return this == other
    if (this.length != other.length) return false

    for (index in 0 until this.length) {
        val thisChar = this[index]
        val otherChar = other[index]
        if (!thisChar.equals(otherChar, ignoreCase)) {
            return false
        }
    }
    return true
}

@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun CharSequence.regionMatches(thisOffset: Int, other: CharSequence, otherOffset: Int, length: Int, ignoreCase: Boolean = false): Boolean =
    regionMatchesImpl(thisOffset, other, otherOffset, length, ignoreCase)

Returns a copy of this string having its first letter titlecased using the rules of the default locale, or the original string if it's empty or already starts with a title case letter.

The title case of a character is usually the same as its upper case with several exceptions.

The particular list of characters with the special title case form depends on the underlying platform.

@sample samples.text.Strings.capitalize
@Deprecated("Use replaceFirstChar instead.", ReplaceWith("replaceFirstChar { if (it.isLowerCase()) it.titlecase() else it.toString() }"))
@DeprecatedSinceKotlin(warningSince = "1.5")
public actual fun String.capitalize(): String {
    return if (isEmpty()) substring(0, 1).uppercase() + substring(1) else this
}

Returns a copy of this string having its first letter lowercased using the rules of the default locale, or the original string if it's empty or already starts with a lower case letter.

@sample samples.text.Strings.decapitalize
@Deprecated("Use replaceFirstChar instead.", ReplaceWith("replaceFirstChar { it.lowercase() }"))
@DeprecatedSinceKotlin(warningSince = "1.5")
public actual fun String.decapitalize(): String {
    return if (isEmpty()) substring(0, 1).lowercase() + substring(1) else this
}

Returns a string containing this char sequence repeated [n] times.

@throws [IllegalArgumentException] when n < 0.

@sample samples.text.Strings.repeat
public actual fun CharSequence.repeat(n: Int): String {
    require(n >= 0) {
        "Count 'n' must be non-negative, but was $n."
    }
    return when (n) {
        0 -> ""
        1 -> this.toString()
        else -> {
            var result = ""
            if (!isEmpty()) {
                var s = this.toString()
                var count = n
                while (true) {
                    if ((count and 1) == 1) {
                        result += s
                    }
                    count = count ushr 1
                    if (count == 0) {
                        break
                    }
                }
                return result
            }
        }
    }
}

Returns a new string obtained by replacing all occurrences of the [oldValue] substring in this string with the specified [newValue] string.

@sample samples.text.Strings.replace
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.replace(oldValue: String, newValue: String, ignoreCase: Boolean = false): String =
    nativeReplace(Regex.escape(oldValue), if (ignoreCase) "gui" else "gu"),

```

```

Regex.nativeEscapeReplacement(newValue))\n\n/**\n * Returns a new string with all occurrences of [oldChar]
replaced with [newChar].\n\n *\n * @sample samples.text.Strings.replace\n
*\n\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n\npublic actual fun
String.replace(oldChar: Char, newChar: Char, ignoreCase: Boolean = false): String =\n
nativeReplace(RegExp(Regex.escape(oldChar.toString()), if (ignoreCase) \"gui\" else \"gu\"),
newChar.toString())\n\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n\npublic actual
fun String.replaceFirst(oldValue: String, newValue: String, ignoreCase: Boolean = false): String =\n
nativeReplace(RegExp(Regex.escape(oldValue), if (ignoreCase) \"ui\" else \"u\"),
Regex.nativeEscapeReplacement(newValue))\n\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGU
MENTS")\n\npublic actual fun String.replaceFirst(oldChar: Char, newChar: Char, ignoreCase: Boolean = false):
String =\n
nativeReplace(RegExp(Regex.escape(oldChar.toString()), if (ignoreCase) \"ui\" else \"u\"),
newChar.toString())\n\n", /*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n\n */\n\npackage kotlin.text\n\n/** Returns the negative [size] if [throwOnMalformed] is
false, throws [CharacterCodingException] otherwise. *\n\nprivate fun malformed(size: Int, index: Int,
throwOnMalformed: Boolean): Int {\n
if (throwOnMalformed) throw CharacterCodingException("Malformed
sequence starting at ${index - 1}")\n
return -size\n}\n\n/**\n * Returns code point corresponding to UTF-16
surrogate pair,\n\n * where the first of the pair is the [high] and the second is in the [string] at the [index].\n\n * Returns
zero if the pair is malformed and [throwOnMalformed] is false.\n\n *\n\n * @throws CharacterCodingException if the
pair is malformed and [throwOnMalformed] is true.\n\n *\n\nprivate fun codePointFromSurrogate(string: String, high:
Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {\n
if (high !in 0xD800..0xDBFF || index >=
endIndex) {\n
return malformed(0, index, throwOnMalformed)\n
}\n
val low = string[index].code\n
if (low !in 0xDC00..0xDFFF) {\n
return malformed(0, index, throwOnMalformed)\n
}\n
return 0x10000 +
((high and 0x3FF) shl 10) or (low and 0x3FF)\n}\n\n/**\n * Returns code point corresponding to UTF-8 sequence of
two bytes,\n\n * where the first byte of the sequence is the [byte1] and the second byte is in the [bytes] array at the
[index].\n\n * Returns zero if the sequence is malformed and [throwOnMalformed] is false.\n\n *\n\n * @throws
CharacterCodingException if the sequence of two bytes is malformed and [throwOnMalformed] is true.\n\n *\n\nprivate fun codePointFrom2(bytes: ByteArray, byte1: Int, index: Int, endIndex: Int, throwOnMalformed:
Boolean): Int {\n
if (byte1 and 0x1E == 0 || index >= endIndex) {\n
return malformed(0, index,
throwOnMalformed)\n
}\n
val byte2 = bytes[index].toInt()\n
if (byte2 and 0xC0 != 0x80) {\n
return
malformed(0, index, throwOnMalformed)\n
}\n
return (byte1 shl 6) xor byte2 xor 0xF80\n}\n\n/**\n * Returns
code point corresponding to UTF-8 sequence of three bytes,\n\n * where the first byte of the sequence is the [byte1]
and the others are in the [bytes] array starting from the [index].\n\n * Returns a non-positive value indicating number
of bytes from [bytes] included in malformed sequence\n\n * if the sequence is malformed and [throwOnMalformed] is
false.\n\n *\n\n * @throws CharacterCodingException if the sequence of three bytes is malformed and
[throwOnMalformed] is true.\n\n *\n\nprivate fun codePointFrom3(bytes: ByteArray, byte1: Int, index: Int, endIndex:
Int, throwOnMalformed: Boolean): Int {\n
if (index >= endIndex) {\n
return malformed(0, index,
throwOnMalformed)\n
}\n\n
val byte2 = bytes[index].toInt()\n
if (byte1 and 0xF == 0) {\n
if (byte2 and
0xE0 != 0xA0) {\n
// Non-shortest form\n
return malformed(0, index, throwOnMalformed)\n
}\n
}\n
else if (byte1 and 0xF == 0xD) {\n
if (byte2 and 0xE0 != 0x80) {\n
// Surrogate code point\n
return malformed(0, index, throwOnMalformed)\n
}\n
}\n
else if (byte2 and 0xC0 != 0x80) {\n
return
malformed(0, index, throwOnMalformed)\n
}\n\n
if (index + 1 == endIndex) {\n
return malformed(1, index,
throwOnMalformed)\n
}\n\n
val byte3 = bytes[index + 1].toInt()\n
if (byte3 and 0xC0 != 0x80) {\n
return
malformed(1, index, throwOnMalformed)\n
}\n\n
return (byte1 shl 12) xor (byte2 shl 6) xor byte3 xor -
0x1E080\n}\n\n/**\n * Returns code point corresponding to UTF-8 sequence of four bytes,\n\n * where the first byte
of the sequence is the [byte1] and the others are in the [bytes] array starting from the [index].\n\n * Returns a non-
positive value indicating number of bytes from [bytes] included in malformed sequence\n\n * if the sequence is
malformed and [throwOnMalformed] is false.\n\n *\n\n * @throws CharacterCodingException if the sequence of four

```



```

bytes is malformed and [throwOnMalformed] is true.\n *\nprivate fun codePointFrom4(bytes: ByteArray, byte1:
Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {\n    if (index >= endIndex) {\n        malformed(0,
index, throwOnMalformed)\n    }\n    val byte2 = bytes[index].toInt()\n    if (byte1 and 0xF == 0x0) {\n        if
(byte2 and 0xF0 <= 0x80) {\n            // Non-shortest form\n            return malformed(0, index,
throwOnMalformed)\n        }\n    } else if (byte1 and 0xF == 0x4) {\n        if (byte2 and 0xF0 != 0x80) {\n            //
Out of Unicode code points domain (larger than U+10FFFF)\n            return malformed(0, index,
throwOnMalformed)\n        }\n    } else if (byte1 and 0xF > 0x4) {\n        return malformed(0, index,
throwOnMalformed)\n    } else if (byte2 and 0xC0 != 0x80) {\n        return malformed(0, index,
throwOnMalformed)\n    }\n    if (index + 1 == endIndex) {\n        return malformed(1, index,
throwOnMalformed)\n    }\n    val byte3 = bytes[index + 1].toInt()\n    if (byte3 and 0xC0 != 0x80) {\n        return
malformed(1, index, throwOnMalformed)\n    }\n    if (index + 2 == endIndex) {\n        return malformed(2, index,
throwOnMalformed)\n    }\n    val byte4 = bytes[index + 2].toInt()\n    if (byte4 and 0xC0 != 0x80) {\n        return
malformed(2, index, throwOnMalformed)\n    }\n    return (byte1 shl 18) xor (byte2 shl 12) xor (byte3 shl 6) xor
byte4 xor 0x381F80)\n}\n\n**\n * Maximum number of bytes needed to encode a single char.\n *\n * Code points in
`0..0x7F` are encoded in a single byte.\n *\n * Code points in `0x80..0x7FFF` are encoded in two bytes.\n *\n * Code points
in `0x800..0xD7FF` or in `0xE000..0xFFFF` are encoded in three bytes.\n *\n * Surrogate code points in
`0xD800..0xDFFF` are not Unicode scalar values, therefore aren't encoded.\n *\n * Code points in
`0x10000..0x10FFFF` are represented by a pair of surrogate `Char`s and are encoded in four bytes.\n *\nprivate
const val MAX_BYTES_PER_CHAR = 3\n\n**\n * The byte sequence a malformed UTF-16 char sequence is
replaced by.\n *\nprivate val REPLACEMENT_BYTE_SEQUENCE: ByteArray = byteArrayOf(0xEF.toByte(),
0xBF.toByte(), 0xBD.toByte())\n\n**\n * Encodes the [string] using UTF-8 and returns the resulting [ByteArray].\n
*\n * @param string the string to encode.\n * @param startIndex the start offset (inclusive) of the substring to
encode.\n * @param endIndex the end offset (exclusive) of the substring to encode.\n * @param
throwOnMalformed whether to throw on malformed char sequence or replace by the
[REPLACEMENT_BYTE_SEQUENCE].\n *\n * @throws CharacterCodingException if the char sequence is
malformed and [throwOnMalformed] is true.\n *\ninternal fun encodeUtf8(string: String, startIndex: Int, endIndex:
Int, throwOnMalformed: Boolean): ByteArray {\n    require(startIndex >= 0 && endIndex <= string.length &&
startIndex <= endIndex)\n    val bytes = ByteArray((endIndex - startIndex) * MAX_BYTES_PER_CHAR)\n    var
byteIndex = 0\n    var charIndex = startIndex\n    while (charIndex < endIndex) {\n        val code =
string[charIndex++].code\n        when {\n            code < 0x80 -> bytes[byteIndex++] = code.toByte()\n
            code < 0x800 -> {\n                bytes[byteIndex++] = ((code shr 6) or 0xC0).toByte()\n
                bytes[byteIndex++] = ((code and 0x3F) or 0x80).toByte()\n            }\n            code < 0xD800 || code >= 0xE000 ->
{\n                bytes[byteIndex++] = ((code shr 12) or 0xE0).toByte()\n                bytes[byteIndex++] = (((code shr 6)
and 0x3F) or 0x80).toByte()\n                bytes[byteIndex++] = ((code and 0x3F) or 0x80).toByte()\n            }\n
            else -> { // Surrogate char value\n                val codePoint = codePointFromSurrogate(string, code, charIndex,
endIndex, throwOnMalformed)\n                if (codePoint <= 0) {\n                    bytes[byteIndex++] =
REPLACEMENT_BYTE_SEQUENCE[0]\n                    bytes[byteIndex++] =
REPLACEMENT_BYTE_SEQUENCE[1]\n                    bytes[byteIndex++] =
REPLACEMENT_BYTE_SEQUENCE[2]\n                } else {\n                    bytes[byteIndex++] = ((codePoint shr
18) or 0xF0).toByte()\n                    bytes[byteIndex++] = (((codePoint shr 12) and 0x3F) or 0x80).toByte()\n
                    bytes[byteIndex++] = (((codePoint shr 6) and 0x3F) or 0x80).toByte()\n                    bytes[byteIndex++] =
(((codePoint and 0x3F) or 0x80).toByte()\n                    charIndex++\n                }\n            }\n        }\n    }
return if (bytes.size == byteIndex) bytes else bytes.copyOf(byteIndex)\n}\n\n**\n * The character a malformed
UTF-8 byte sequence is replaced by.\n *\nprivate const val REPLACEMENT_CHAR = "\uFFFF"\n\n**\n *
Decodes the UTF-8 [bytes] array and returns the resulting [String].\n *\n * @param bytes the byte array to decode.\n
* @param startIndex the start offset (inclusive) of the array to be decoded.\n * @param endIndex the end offset
(exclusive) of the array to be encoded.\n * @param throwOnMalformed whether to throw on malformed byte
sequence or replace by the [REPLACEMENT_CHAR].\n *\n * @throws CharacterCodingException if the array is

```

```

malformed UTF-8 byte sequence and [throwOnMalformed] is true.\n *\ninternal fun decodeUtf8(bytes: ByteArray,
startIndex: Int, endIndex: Int, throwOnMalformed: Boolean): String {\n    require(startIndex >= 0 && endIndex <=
bytes.size && startIndex <= endIndex)\n    var byteIndex = startIndex\n    val stringBuilder = StringBuilder()\n    while (byteIndex < endIndex) {\n        val byte = bytes[byteIndex++].toInt()\n        when {\n            byte >= 0 ->\n                stringBuilder.append(byte.toChar())\n            byte shr 5 == -2 -> {\n                val code =
codePointFrom2(bytes, byte, byteIndex, endIndex, throwOnMalformed)\n                if (code <= 0) {\n
stringBuilder.append(REPLACEMENT_CHAR)\n                byteIndex += -code\n                } else {\n
stringBuilder.append(code.toChar())\n                byteIndex += 1\n                }\n            }\n            byte shr 4 == -2 -
> {\n                val code = codePointFrom3(bytes, byte, byteIndex, endIndex, throwOnMalformed)\n                if
(code <= 0) {\n                stringBuilder.append(REPLACEMENT_CHAR)\n                byteIndex += -code\n
                } else {\n                stringBuilder.append(code.toChar())\n                byteIndex += 2\n                }\n
            }\n            byte shr 3 == -2 -> {\n                val code = codePointFrom4(bytes, byte, byteIndex, endIndex,
throwOnMalformed)\n                if (code <= 0) {\n                stringBuilder.append(REPLACEMENT_CHAR)\n
                byteIndex += -code\n                } else {\n                val high = (code - 0x10000) shr 10 or 0xD800\n
                val low = (code and 0x3FF) or 0xDC00\n                stringBuilder.append(high.toChar())\n
stringBuilder.append(low.toChar())\n                byteIndex += 3\n                }\n            }\n            else -> {\n
                malformed(0, byteIndex, throwOnMalformed)\n                stringBuilder.append(REPLACEMENT_CHAR)\n
            }\n        }\n    }\n    return stringBuilder.toString()\n}"/**\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n *\npackage kotlin\n\n/**\n * Returns the detailed description of this
throwable with its stack trace.\n *\n * The detailed description includes:\n * - the short description (see
[Throwable.toString]) of this throwable;\n * - the complete stack trace;\n * - detailed descriptions of the exceptions
that were [suppressed][suppressedExceptions] in order to deliver this exception;\n * - the detailed description of
each throwable in the [Throwable.cause] chain.\n *\n@SinceKotlin("1.4")\npublic actual fun
Throwable.stackTraceToString(): String = ExceptionTraceBuilder().buildFor(this)\n\n/**\n * Prints the [detailed
description][Throwable.stackTraceToString] of this throwable to console error output.\n
*\n@SinceKotlin("1.4")\npublic actual fun Throwable.printStackTrace() {\n
console.error(this.stackTraceToString())\n}\n\n/**\n * Adds the specified exception to the list of exceptions that
were\n * suppressed in order to deliver this exception.\n *\n@SinceKotlin("1.4")\npublic actual fun
Throwable.addSuppressed(exception: Throwable) {\n    if (this !== exception) {\n        val suppressed =
this.asDynamic()._suppressed.unsafeCast<MutableList<Throwable?>>()\n        if (suppressed == null) {\n
this.asDynamic()._suppressed = mutableListOf(exception)\n        } else {\n            suppressed.add(exception)\n
        }\n    }\n}\n\n/**\n * Returns a list of all exceptions that were suppressed in order to deliver this exception.\n
*\n@SinceKotlin("1.4")\npublic actual val Throwable.suppressedExceptions: List<Throwable>\n    get() {\n
return this.asDynamic()._suppressed?.unsafeCast<List<Throwable>>() ?: emptyList()\n    }\n\nprivate class
ExceptionTraceBuilder {\n    private val target = StringBuilder()\n    private val visited = arrayOf<Throwable>()\n
private var topStack: String = ""\n    private var topStackStart: Int = 0\n    fun buildFor(exception: Throwable):
String {\n        exception.dumpFullTrace("", "")\n        return target.toString()\n    }\n    private fun
hasSeen(exception: Throwable): Boolean = visited.any { it === exception }\n    private fun
Throwable.dumpFullTrace(indent: String, qualifier: String) {\n        this.dumpSelfTrace(indent, qualifier) ||
return\n        var cause = this.cause\n        while (cause != null) {\n            cause.dumpSelfTrace(indent, "Caused
by: ") || return\n            cause = cause.cause\n        }\n    }\n    private fun Throwable.dumpSelfTrace(indent:
String, qualifier: String): Boolean {\n        target.append(indent).append(qualifier)\n        val shortInfo =
this.toString()\n        if (hasSeen(this)) {\n            target.append("[CIRCULAR REFERENCE, SEE ABOVE:
").append(shortInfo).append("\\n")\n            return false\n        }\n        visited.asDynamic().push(this)\n        var
stack = this.asDynamic().stack as String?\n        if (stack != null) {\n            val stackStart =
stack.indexOf(shortInfo).let { if (it < 0) 0 else it + shortInfo.length }\n            if (stackStart == 0)
target.append(shortInfo).append("\\n")\n            if (topStack.isEmpty()) {\n                topStack = stack\n

```

```

topStackStart = stackStart\n        } else {\n            stack = dropCommonFrames(stack, stackStart)\n        }\n        if (indent.isNotEmpty()) {\n            // indent stack, but avoid indenting exception message lines\n            val\n            messageLines = if (stackStart == 0) 0 else 1 + shortInfo.count { c -> c == '\n' }\n            stack.lineSequence().forEachIndexed { index: Int, line: String ->\n                if (index >= messageLines)\n                target.append(indent)\n                    target.append(line).append("\n")\n                }\n            } else {\n            target.append(stack).append("\n")\n                }\n            } else {\n            target.append(shortInfo).append("\n")\n        }\n        val suppressed = suppressedExceptions\n        if (suppressed.isNotEmpty()) {\n            val\n            suppressedIndent = indent + "\n\n" for (s in suppressed) {\n                s.dumpFullTrace(suppressedIndent,\n                \"Suppressed: \")\n            }\n            return true\n        }\n        private fun dropCommonFrames(stack: String,\n        stackStart: Int): String {\n            var commonFrames: Int = 0\n            var lastBreak: Int = 0\n            var preLastBreak: Int\n            = 0\n            for (pos in 0 until minOf(topStack.length - topStackStart, stack.length - stackStart)) {\n                val c =\n                stack[stack.lastIndex - pos]\n                if (c != topStack[topStack.lastIndex - pos]) break\n                if (c == '\n') {\n                    commonFrames += 1\n                    preLastBreak = lastBreak\n                    lastBreak = pos\n                }\n            }\n            if (commonFrames <= 1) return stack\n                while (preLastBreak > 0 && stack[stack.lastIndex - (preLastBreak - 1)]\n                == '\n')\n                preLastBreak -= 1\n                // leave 1 common frame to ease matching with the top exception stack\n                return stack.dropLast(preLastBreak) + \"... and ${commonFrames - 1} more common stack frames skipped\"\n            }\n        }\n        /*\n        * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n        * Use of this\n        source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n        */\n        package kotlin.time\n        import kotlin.js.json\n        import kotlin.math.*\n        internal actual inline val\n        durationAssertionsEnabled: Boolean get() = true\n        internal actual fun formatToExactDecimals(value: Double,\n        decimals: Int): String {\n            val rounded = if (decimals == 0) {\n                value\n            } else {\n                val pow =\n                10.0.pow(decimals)\n                JsMath.round(abs(value) * pow) / pow * sign(value)\n            }\n            return if (abs(rounded) <\n            1e21) {\n                // toFixed switches to scientific format after 1e21\n                rounded.asDynamic().toFixed(decimals).unsafeCast<String>()\n            } else {\n                // toPrecision outputs the specified\n                number of digits, but only for positive numbers\n                val positive = abs(rounded)\n                val positiveString =\n                positive.asDynamic().toPrecision(ceil(log10(positive)) + decimals).unsafeCast<String>()\n                if (rounded < 0) \"-\n                $positiveString\"\n                else positiveString\n            }\n        }\n        internal actual fun formatUpToDecimals(value: Double, decimals:\n        Int): String {\n            return value.asDynamic().toLocaleString(\"en-us\", json(\"maximumFractionDigits\" to\n            decimals)).unsafeCast<String>()\n        }\n        /*\n        * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming\n        Language contributors.\n        * Use of this source code is governed by the Apache 2.0 license that can be found in the\n        license/LICENSE.txt file.\n        */\n        package\n        kotlin.time\n        @SinceKotlin(\"1.6\")\n        @WasExperimental(ExperimentalTime::class)\n        public actual enum class\n        DurationUnit(internal val scale: Double) {\n            /**\n            * Time unit representing one nanosecond, which is 1/1000 of\n            a microsecond.\n            */\n            NANOSECONDS(1e0),\n            /**\n            * Time unit representing one microsecond, which is\n            1/1000 of a millisecond.\n            */\n            MICROSECONDS(1e3),\n            /**\n            * Time unit representing one millisecond,\n            which is 1/1000 of a second.\n            */\n            MILLISECONDS(1e6),\n            /**\n            * Time unit representing one second.\n            */\n            SECONDS(1e9),\n            /**\n            * Time unit representing one minute.\n            */\n            MINUTES(60e9),\n            /**\n            * Time unit representing one hour.\n            */\n            HOURS(3600e9),\n            /**\n            * Time unit representing one day,\n            which is always equal to 24 hours.\n            */\n            DAYS(86400e9);\n        }\n        @SinceKotlin(\"1.3\")\n        internal actual fun\n        convertDurationUnit(value: Double, sourceUnit: DurationUnit, targetUnit: DurationUnit): Double {\n            val\n            sourceCompareTarget = sourceUnit.scale.compareTo(targetUnit.scale)\n            return when {\n            sourceCompareTarget > 0 -> value * (sourceUnit.scale / targetUnit.scale)\n            sourceCompareTarget < 0 -> value /\n            (targetUnit.scale / sourceUnit.scale)\n            else -> value\n            }\n        }\n        @SinceKotlin(\"1.5\")\n        internal actual fun\n        convertDurationUnitOverflow(value: Long, sourceUnit: DurationUnit, targetUnit: DurationUnit): Long {\n            val\n            sourceCompareTarget = sourceUnit.scale.compareTo(targetUnit.scale)\n            return when {\n            sourceCompareTarget > 0 -> value * (sourceUnit.scale / targetUnit.scale).toLong()\n            sourceCompareTarget < 0\n            -> value / (targetUnit.scale / sourceUnit.scale).toLong()\n            else -> value\n            }\n        }\n        @SinceKotlin(\"1.5\")\n        internal actual fun convertDurationUnit(value: Long, sourceUnit: DurationUnit,

```

```

targetUnit: DurationUnit): Long {\n    val sourceCompareTarget = sourceUnit.scale.compareTo(targetUnit.scale)\n
return when {\n    sourceCompareTarget > 0 -> {\n        val scale = (sourceUnit.scale /\n
targetUnit.scale).toLong()\n        val result = value * scale\n        when {\n            result / scale == value ->\n
result\n            value > 0 -> Long.MAX_VALUE\n            else -> Long.MIN_VALUE\n        }\n    }\n
sourceCompareTarget < 0 -> value / (targetUnit.scale / sourceUnit.scale).toLong()\n    else -> value\n
}\n}\n\n\n", /*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of\n
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\npackage kotlin.time\nimport org.w3c.performance.GlobalPerformance\nimport\n
org.w3c.performance.Performance\nimport kotlin.math.truncate\nimport\n
kotlin.time.Duration.Companion.milliseconds\nimport\n
kotlin.time.TimeSource.Monotonic.ValueTimeMark\n\n@Suppress(\"ACTUAL_WITHOUT_EXPECT\") //\n
visibility\ninternal actual typealias ValueTimeMarkReading = Any\n\n@ExperimentalTime\ninternal interface\n
DefaultTimeSource : TimeSource {\n    override fun markNow(): ValueTimeMark\n    fun elapsedFrom(timeMark:\n
ValueTimeMark): Duration\n    fun adjustReading(timeMark: ValueTimeMark, duration: Duration):\n
ValueTimeMark\n}\n\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\ninternal actual object MonotonicTimeSource\n
: DefaultTimeSource, TimeSource { // TODO: interface should not be required here\n    private val actualSource:\n
DefaultTimeSource = run {\n        val isNode: Boolean = js(\"typeof process !== 'undefined' && process.versions\n
&& !process.versions.node\")\n        if (isNode)\n            HrTimeSource(js(\"process\").unsafeCast<Process>())\n
        else\n            js(\"typeof self !== 'undefined' ? self : globalThis\")\n
    }.unsafeCast<GlobalPerformance?>()\n        ?.performance\n        ?.let(::PerformanceTimeSource)\n
        ?: DateNowTimeSource\n    }\n    actual override fun markNow(): ValueTimeMark = actualSource.markNow()\n
    actual override fun elapsedFrom(timeMark: ValueTimeMark): Duration = actualSource.elapsedFrom(timeMark)\n
    actual override fun adjustReading(timeMark: ValueTimeMark, duration: Duration): ValueTimeMark =\n
actualSource.adjustReading(timeMark, duration)\n}\n\ninternal external interface Process {\n    fun hrtime(time:\n
Array<Double> = definedExternally): Array<Double>}\n\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\ninternal\n
class HrTimeSource(private val process: Process) : DefaultTimeSource {\n    override fun markNow():\n
ValueTimeMark = ValueTimeMark(process.hrtime())\n    override fun elapsedFrom(timeMark: ValueTimeMark):\n
Duration =\n        @Suppress(\"UNCHECKED_CAST\")\n        process.hrtime(timeMark.reading as\n
Array<Double>)\n        .let { (seconds, nanos) -> seconds.toDuration(DurationUnit.SECONDS) +\n
nanos.toDuration(DurationUnit.NANOSECONDS) }\n    override fun adjustReading(timeMark: ValueTimeMark,\n
duration: Duration): ValueTimeMark =\n        @Suppress(\"UNCHECKED_CAST\")\n        (timeMark.reading as\n
Array<Double>).let { (seconds, nanos) ->\n            duration.toComponents { _, addNanos ->\n
arrayOf<Double>(sumCheckNaN(seconds + truncate(duration.toDouble(DurationUnit.SECONDS))), nanos +\n
addNanos)\n        }\n        }.let(TimeSource.Monotonic::ValueTimeMark)\n}\n\n    override fun toString(): String =\n
\"TimeSource(process.hrtime())\"\n}\n\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\ninternal class\n
PerformanceTimeSource(val performance: Performance) : DefaultTimeSource { //\n
AbstractDoubleTimeSource(unit = DurationUnit.MILLISECONDS) {\n    private fun read(): Double =\n
performance.now()\n    override fun markNow(): ValueTimeMark = ValueTimeMark(read())\n    override fun\n
elapsedFrom(timeMark: ValueTimeMark): Duration = (read() - timeMark.reading as Double).milliseconds\n
    override fun adjustReading(timeMark: ValueTimeMark, duration: Duration): ValueTimeMark =\n
ValueTimeMark(sumCheckNaN(timeMark.reading as Double +\n
duration.toDouble(DurationUnit.MILLISECONDS)))\n    override fun toString(): String =\n
\"TimeSource(self.performance.now())\"\n}\n\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\ninternal object\n
DateNowTimeSource : DefaultTimeSource {\n    private fun read(): Double = kotlin.js.Date.now()\n    override\n
fun markNow(): ValueTimeMark = ValueTimeMark(read())\n    override fun elapsedFrom(timeMark:\n
ValueTimeMark): Duration = (read() - timeMark.reading as Double).milliseconds\n    override fun\n
adjustReading(timeMark: ValueTimeMark, duration: Duration): ValueTimeMark =\n
ValueTimeMark(sumCheckNaN(timeMark.reading as Double +

```



```

removeChild(firstChild!!)\n    }\n}\n\n/**\n * Creates text node and append it to the element.\n *\n * @return this element\n *\n * @Since Kotlin("1.4")\nfun Element.appendText(text: String): Element {\n    appendChild(ownerDocument!!.createTextNode(text))\n    return this\n}\n\n"/**\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n * \npackage org.w3c.dom\n\n@Deprecated("Use UnionMessagePortOrWindowProxy instead.", ReplaceWith("UnionMessagePortOrWindowProxy"))\ntypealias UnionMessagePortOrWindow = UnionMessagePortOrWindowProxy\n\n@Deprecated("Use `as` instead.", ReplaceWith("`as`"))\nvar HTMLLinkElement.as_\n    get() = `as`\n    set(value) {\n        `as` = value\n    }\n\n@Deprecated("Use `is` instead.", ReplaceWith("`is`"))\nvar ElementCreationOptions.is_\n    get() = `is`\n    set(value) {\n        `is` = value\n    }"/**\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n * \n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage org.khronos.webgl\n\nimport kotlin.js.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\n\npublic external interface WebGLContextAttributes {\n    var alpha: Boolean? /* = true */\n        get() = definedExternally\n        set(value) = definedExternally\n    var depth: Boolean? /* = true */\n        get() = definedExternally\n        set(value) = definedExternally\n    var stencil: Boolean? /* = false */\n        get() = definedExternally\n        set(value) = definedExternally\n    var antialias: Boolean? /* = true */\n        get() = definedExternally\n        set(value) = definedExternally\n    var premultipliedAlpha: Boolean? /* = true */\n        get() = definedExternally\n        set(value) = definedExternally\n    var preserveDrawingBuffer: Boolean? /* = false */\n        get() = definedExternally\n        set(value) = definedExternally\n    var preferLowPowerToHighPerformance: Boolean? /* = false */\n        get() = definedExternally\n        set(value) = definedExternally\n    var failIfMajorPerformanceCaveat: Boolean? /* = false */\n        get() = definedExternally\n        set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun WebGLContextAttributes(alpha: Boolean? = true, depth: Boolean? = true, stencil: Boolean? = false, antialias: Boolean? = true, premultipliedAlpha: Boolean? = true, preserveDrawingBuffer: Boolean? = false, preferLowPowerToHighPerformance: Boolean? = false, failIfMajorPerformanceCaveat: Boolean? = false): WebGLContextAttributes {\n    val o = js("{}")\n    o["alpha"] = alpha\n    o["depth"] = depth\n    o["stencil"] = stencil\n    o["antialias"] = antialias\n    o["premultipliedAlpha"] = premultipliedAlpha\n    o["preserveDrawingBuffer"] = preserveDrawingBuffer\n    o["preferLowPowerToHighPerformance"] = preferLowPowerToHighPerformance\n    o["failIfMajorPerformanceCaveat"] = failIfMajorPerformanceCaveat\n    return o\n}\n\npublic external abstract class WebGLObject\n\n/**\n * Exposes the JavaScript [WebGLBuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLBuffer) to Kotlin\n *\n * \npublic external abstract class WebGLBuffer : WebGLObject\n\n/**\n * Exposes the JavaScript [WebGLFramebuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLFramebuffer) to Kotlin\n *\n * \npublic external abstract class WebGLFramebuffer : WebGLObject\n\n/**\n * Exposes the JavaScript [WebGLProgram](https://developer.mozilla.org/en/docs/Web/API/WebGLProgram) to Kotlin\n *\n * \npublic external abstract class WebGLProgram : WebGLObject\n\n/**\n * Exposes the JavaScript [WebGLRenderbuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLRenderbuffer) to Kotlin\n *\n * \npublic external abstract class WebGLRenderbuffer : WebGLObject\n\n/**\n * Exposes the JavaScript [WebGLShader](https://developer.mozilla.org/en/docs/Web/API/WebGLShader) to Kotlin\n *\n * \npublic external abstract class WebGLShader : WebGLObject\n\n/**\n * Exposes the JavaScript [WebGLTexture](https://developer.mozilla.org/en/docs/Web/API/WebGLTexture) to Kotlin\n *\n * \npublic external abstract class WebGLTexture : WebGLObject\n\n/**\n * Exposes the JavaScript [WebGLUniformLocation](https://developer.mozilla.org/en/docs/Web/API/WebGLUniformLocation) to Kotlin\n *\n * \npublic external abstract class WebGLUniformLocation\n\n/**\n * Exposes the JavaScript [WebGLActiveInfo](https://developer.mozilla.org/en/docs/Web/API/WebGLActiveInfo) to Kotlin\n *\n * \npublic external abstract class WebGLActiveInfo {\n    open val size: Int\n    open val type: Int\n    open val name:

```

String}\n\n/**\n * Exposes the JavaScript

[WebGLShaderPrecisionFormat](https://developer.mozilla.org/en/docs/Web/API/WebGLShaderPrecisionFormat) to

Kotlin\n *\npublic external abstract class WebGLShaderPrecisionFormat {\n open val rangeMin: Int\n open val rangeMax: Int\n open val precision:

```
Int\n}\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface WebGLRenderingContextBase {\n    val canvas: HTMLCanvasElement\n    val drawingBufferWidth: Int\n    val drawingBufferHeight: Int\n    fun getContextAttributes(): WebGLContextAttributes?\n    fun isContextLost(): Boolean\n    fun getSupportedExtensions(): Array<String>?\n    fun getExtension(name: String): dynamic\n    fun activeTexture(texture: Int)\n    fun attachShader(program: WebGLProgram?, shader: WebGLShader?)\n    fun bindAttribLocation(program: WebGLProgram?, index: Int, name: String)\n    fun bindBuffer(target: Int, buffer: WebGLBuffer?)\n    fun bindFramebuffer(target: Int, framebuffer: WebGLFramebuffer?)\n    fun bindRenderbuffer(target: Int, renderbuffer: WebGLRenderbuffer?)\n    fun bindTexture(target: Int, texture: WebGLTexture?)\n    fun blendColor(red: Float, green: Float, blue: Float, alpha: Float)\n    fun blendEquation(mode: Int)\n    fun blendEquationSeparate(modeRGB: Int, modeAlpha: Int)\n    fun blendFunc(sfactor: Int, dfactor: Int)\n    fun blendFuncSeparate(srcRGB: Int, dstRGB: Int, srcAlpha: Int, dstAlpha: Int)\n    fun bufferData(target: Int, size: Int, usage: Int)\n    fun bufferData(target: Int, data: BufferDataSource?, usage: Int)\n    fun bufferSubData(target: Int, offset: Int, data: BufferDataSource?)\n    fun checkFramebufferStatus(target: Int): Int\n    fun clear(mask: Int)\n    fun clearColor(red: Float, green: Float, blue: Float, alpha: Float)\n    fun clearDepth(depth: Float)\n    fun clearStencil(s: Int)\n    fun colorMask(red: Boolean, green: Boolean, blue: Boolean, alpha: Boolean)\n    fun compileShader(shader: WebGLShader?)\n    fun compressedTexImage2D(target: Int, level: Int, internalformat: Int, width: Int, height: Int, border: Int, data: ArrayBufferView)\n    fun compressedTexSubImage2D(target: Int, level: Int, xoffset: Int, yoffset: Int, width: Int, height: Int, format: Int, data: ArrayBufferView)\n    fun copyTexImage2D(target: Int, level: Int, internalformat: Int, x: Int, y: Int, width: Int, height: Int, border: Int)\n    fun copyTexSubImage2D(target: Int, level: Int, xoffset: Int, yoffset: Int, x: Int, y: Int, width: Int, height: Int)\n    fun createBuffer(): WebGLBuffer?\n    fun createFramebuffer(): WebGLFramebuffer?\n    fun createProgram(): WebGLProgram?\n    fun createRenderbuffer(): WebGLRenderbuffer?\n    fun createShader(type: Int): WebGLShader?\n    fun createTexture(): WebGLTexture?\n    fun cullFace(mode: Int)\n    fun deleteBuffer(buffer: WebGLBuffer?)\n    fun deleteFramebuffer(framebuffer: WebGLFramebuffer?)\n    fun deleteProgram(program: WebGLProgram?)\n    fun deleteRenderbuffer(renderbuffer: WebGLRenderbuffer?)\n    fun deleteShader(shader: WebGLShader?)\n    fun deleteTexture(texture: WebGLTexture?)\n    fun depthFunc(func: Int)\n    fun depthMask(flag: Boolean)\n    fun depthRange(zNear: Float, zFar: Float)\n    fun detachShader(program: WebGLProgram?, shader: WebGLShader?)\n    fun disable(cap: Int)\n    fun disableVertexAttribArray(index: Int)\n    fun drawArrays(mode: Int, first: Int, count: Int)\n    fun drawElements(mode: Int, count: Int, type: Int, offset: Int)\n    fun enable(cap: Int)\n    fun enableVertexAttribArray(index: Int)\n    fun finish()\n    fun flush()\n    fun framebufferRenderbuffer(target: Int, attachment: Int, renderbuffertarget: Int, renderbuffer: WebGLRenderbuffer?)\n    fun framebufferTexture2D(target: Int, attachment: Int, textarget: Int, texture: WebGLTexture?, level: Int)\n    fun frontFace(mode: Int)\n    fun generateMipmap(target: Int)\n    fun getActiveAttrib(program: WebGLProgram?, index: Int): WebGLActiveInfo?\n    fun getActiveUniform(program: WebGLProgram?, index: Int): WebGLActiveInfo?\n    fun getAttachedShaders(program: WebGLProgram?): Array<WebGLShader>?\n    fun getAttribLocation(program: WebGLProgram?, name: String): Int\n    fun getBufferParameter(target: Int, pname: Int): Any?\n    fun getParameter(pname: Int): Any?\n    fun getError(): Int\n    fun getFramebufferAttachmentParameter(target: Int, attachment: Int, pname: Int): Any?\n    fun getProgramParameter(program: WebGLProgram?, pname: Int): Any?\n    fun getProgramInfoLog(program: WebGLProgram?): String?\n    fun getRenderbufferParameter(target: Int, pname: Int): Any?\n    fun getShaderParameter(shader: WebGLShader?, pname: Int): Any?\n    fun getShaderPrecisionFormat(shadertype: Int, precisiontype: Int): WebGLShaderPrecisionFormat?\n    fun getShaderInfoLog(shader: WebGLShader?): String?\n    fun getShaderSource(shader: WebGLShader?): String?\n    fun getTexParameter(target: Int, pname: Int): Any?\n    fun getUniform(program: WebGLProgram?, location:
```

```

WebGLUniformLocation?): Any?\n fun getLocation(program: WebGLProgram?, name: String):
WebGLUniformLocation?\n fun getVertexAttrib(index: Int, pname: Int): Any?\n fun
getVertexAttribOffset(index: Int, pname: Int): Int\n fun hint(target: Int, mode: Int)\n fun isBuffer(buffer:
WebGLBuffer?): Boolean\n fun isEnabled(cap: Int): Boolean\n fun isFramebuffer( framebuffer:
WebGLFramebuffer?): Boolean\n fun isProgram(program: WebGLProgram?): Boolean\n fun
isRenderbuffer(renderbuffer: WebGLRenderbuffer?): Boolean\n fun isShader(shader: WebGLShader?): Boolean\n
fun isTexture(texture: WebGLTexture?): Boolean\n fun lineWidth(width: Float)\n fun linkProgram(program:
WebGLProgram?)\n fun pixelStorei(pname: Int, param: Int)\n fun polygonOffset(factor: Float, units: Float)\n
fun readPixels(x: Int, y: Int, width: Int, height: Int, format: Int, type: Int, pixels: ArrayBufferView?)\n fun
renderbufferStorage(target: Int, internalformat: Int, width: Int, height: Int)\n fun sampleCoverage(value: Float,
invert: Boolean)\n fun scissor(x: Int, y: Int, width: Int, height: Int)\n fun shaderSource(shader: WebGLShader?,
source: String)\n fun stencilFunc(func: Int, ref: Int, mask: Int)\n fun stencilFuncSeparate(face: Int, func: Int, ref:
Int, mask: Int)\n fun stencilMask(mask: Int)\n fun stencilMaskSeparate(face: Int, mask: Int)\n fun
stencilOp(fail: Int, zfail: Int, zpass: Int)\n fun stencilOpSeparate(face: Int, fail: Int, zfail: Int, zpass: Int)\n fun
texImage2D(target: Int, level: Int, internalformat: Int, width: Int, height: Int, border: Int, format: Int, type: Int, pixels:
ArrayBufferView?)\n fun texImage2D(target: Int, level: Int, internalformat: Int, format: Int, type: Int, source:
TexImageSource?)\n fun texParameterf(target: Int, pname: Int, param: Float)\n fun texParameteri(target: Int,
pname: Int, param: Int)\n fun texSubImage2D(target: Int, level: Int, xoffset: Int, yoffset: Int, width: Int, height: Int,
format: Int, type: Int, pixels: ArrayBufferView?)\n fun texSubImage2D(target: Int, level: Int, xoffset: Int, yoffset:
Int, format: Int, type: Int, source: TexImageSource?)\n fun uniform1f(location: WebGLUniformLocation?, x:
Float)\n fun uniform1fv(location: WebGLUniformLocation?, v: Float32Array)\n fun uniform1fv(location:
WebGLUniformLocation?, v: Array<Float>)\n fun uniform1i(location: WebGLUniformLocation?, x: Int)\n fun
uniform1iv(location: WebGLUniformLocation?, v: Int32Array)\n fun uniform1iv(location:
WebGLUniformLocation?, v: Array<Int>)\n fun uniform2f(location: WebGLUniformLocation?, x: Float, y:
Float)\n fun uniform2fv(location: WebGLUniformLocation?, v: Float32Array)\n fun uniform2fv(location:
WebGLUniformLocation?, v: Array<Float>)\n fun uniform2i(location: WebGLUniformLocation?, x: Int, y: Int)\n
fun uniform2iv(location: WebGLUniformLocation?, v: Int32Array)\n fun uniform2iv(location:
WebGLUniformLocation?, v: Array<Int>)\n fun uniform3f(location: WebGLUniformLocation?, x: Float, y: Float,
z: Float)\n fun uniform3fv(location: WebGLUniformLocation?, v: Float32Array)\n fun uniform3fv(location:
WebGLUniformLocation?, v: Array<Float>)\n fun uniform3i(location: WebGLUniformLocation?, x: Int, y: Int, z:
Int)\n fun uniform3iv(location: WebGLUniformLocation?, v: Int32Array)\n fun uniform3iv(location:
WebGLUniformLocation?, v: Array<Int>)\n fun uniform4f(location: WebGLUniformLocation?, x: Float, y: Float,
z: Float, w: Float)\n fun uniform4fv(location: WebGLUniformLocation?, v: Float32Array)\n fun
uniform4fv(location: WebGLUniformLocation?, v: Array<Float>)\n fun uniform4i(location:
WebGLUniformLocation?, x: Int, y: Int, z: Int, w: Int)\n fun uniform4iv(location: WebGLUniformLocation?, v:
Int32Array)\n fun uniform4iv(location: WebGLUniformLocation?, v: Array<Int>)\n fun
uniformMatrix2fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)\n fun
uniformMatrix2fv(location: WebGLUniformLocation?, transpose: Boolean, value: Array<Float>)\n fun
uniformMatrix3fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)\n fun
uniformMatrix3fv(location: WebGLUniformLocation?, transpose: Boolean, value: Array<Float>)\n fun
uniformMatrix4fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)\n fun
uniformMatrix4fv(location: WebGLUniformLocation?, transpose: Boolean, value: Array<Float>)\n fun
useProgram(program: WebGLProgram?)\n fun validateProgram(program: WebGLProgram?)\n fun
vertexAttrib1f(index: Int, x: Float)\n fun vertexAttrib1fv(index: Int, values: dynamic)\n fun
vertexAttrib2f(index: Int, x: Float, y: Float)\n fun vertexAttrib2fv(index: Int, values: dynamic)\n fun
vertexAttrib3f(index: Int, x: Float, y: Float, z: Float)\n fun vertexAttrib3fv(index: Int, values: dynamic)\n fun
vertexAttrib4f(index: Int, x: Float, y: Float, z: Float, w: Float)\n fun vertexAttrib4fv(index: Int, values: dynamic)\n
fun vertexAttribPointer(index: Int, size: Int, type: Int, normalized: Boolean, stride: Int, offset: Int)\n fun

```



```

viewport(x: Int, y: Int, width: Int, height: Int)\n\n companion object {\n    val DEPTH_BUFFER_BIT: Int\n    val STENCIL_BUFFER_BIT: Int\n    val COLOR_BUFFER_BIT: Int\n    val POINTS: Int\n    val LINES: Int\n    val LINE_LOOP: Int\n    val LINE_STRIP: Int\n    val TRIANGLES: Int\n    val TRIANGLE_STRIP: Int\n    val TRIANGLE_FAN: Int\n    val ZERO: Int\n    val ONE: Int\n    val SRC_COLOR: Int\n    val ONE_MINUS_SRC_COLOR: Int\n    val SRC_ALPHA: Int\n    val ONE_MINUS_SRC_ALPHA: Int\n    val DST_ALPHA: Int\n    val ONE_MINUS_DST_ALPHA: Int\n    val DST_COLOR: Int\n    val ONE_MINUS_DST_COLOR: Int\n    val SRC_ALPHA_SATURATE: Int\n    val FUNC_ADD: Int\n    val BLEND_EQUATION: Int\n    val BLEND_EQUATION_RGB: Int\n    val BLEND_EQUATION_ALPHA: Int\n    val FUNC_SUBTRACT: Int\n    val FUNC_REVERSE_SUBTRACT: Int\n    val BLEND_DST_RGB: Int\n    val BLEND_SRC_RGB: Int\n    val BLEND_DST_ALPHA: Int\n    val BLEND_SRC_ALPHA: Int\n    val CONSTANT_COLOR: Int\n    val ONE_MINUS_CONSTANT_COLOR: Int\n    val CONSTANT_ALPHA: Int\n    val ONE_MINUS_CONSTANT_ALPHA: Int\n    val BLEND_COLOR: Int\n    val ARRAY_BUFFER: Int\n    val ELEMENT_ARRAY_BUFFER: Int\n    val ARRAY_BUFFER_BINDING: Int\n    val ELEMENT_ARRAY_BUFFER_BINDING: Int\n    val STREAM_DRAW: Int\n    val STATIC_DRAW: Int\n    val DYNAMIC_DRAW: Int\n    val BUFFER_SIZE: Int\n    val BUFFER_USAGE: Int\n    val CURRENT_VERTEX_ATTRIB: Int\n    val FRONT: Int\n    val BACK: Int\n    val FRONT_AND_BACK: Int\n    val CULL_FACE: Int\n    val BLEND: Int\n    val DITHER: Int\n    val STENCIL_TEST: Int\n    val DEPTH_TEST: Int\n    val SCISSOR_TEST: Int\n    val POLYGON_OFFSET_FILL: Int\n    val SAMPLE_ALPHA_TO_COVERAGE: Int\n    val SAMPLE_COVERAGE: Int\n    val NO_ERROR: Int\n    val INVALID_ENUM: Int\n    val INVALID_VALUE: Int\n    val INVALID_OPERATION: Int\n    val OUT_OF_MEMORY: Int\n    val CW: Int\n    val CCW: Int\n    val LINE_WIDTH: Int\n    val ALIASED_POINT_SIZE_RANGE: Int\n    val ALIASED_LINE_WIDTH_RANGE: Int\n    val CULL_FACE_MODE: Int\n    val FRONT_FACE: Int\n    val DEPTH_RANGE: Int\n    val DEPTH_WRITEMASK: Int\n    val DEPTH_CLEAR_VALUE: Int\n    val DEPTH_FUNC: Int\n    val STENCIL_CLEAR_VALUE: Int\n    val STENCIL_FUNC: Int\n    val STENCIL_FAIL: Int\n    val STENCIL_PASS_DEPTH_FAIL: Int\n    val STENCIL_PASS_DEPTH_PASS: Int\n    val STENCIL_REF: Int\n    val STENCIL_VALUE_MASK: Int\n    val STENCIL_WRITEMASK: Int\n    val STENCIL_BACK_FUNC: Int\n    val STENCIL_BACK_FAIL: Int\n    val STENCIL_BACK_PASS_DEPTH_FAIL: Int\n    val STENCIL_BACK_PASS_DEPTH_PASS: Int\n    val STENCIL_BACK_REF: Int\n    val STENCIL_BACK_VALUE_MASK: Int\n    val STENCIL_BACK_WRITEMASK: Int\n    val VIEWPORT: Int\n    val SCISSOR_BOX: Int\n    val COLOR_CLEAR_VALUE: Int\n    val COLOR_WRITEMASK: Int\n    val UNPACK_ALIGNMENT: Int\n    val PACK_ALIGNMENT: Int\n    val MAX_TEXTURE_SIZE: Int\n    val MAX_VIEWPORT_DIMS: Int\n    val SUBPIXEL_BITS: Int\n    val RED_BITS: Int\n    val GREEN_BITS: Int\n    val BLUE_BITS: Int\n    val ALPHA_BITS: Int\n    val DEPTH_BITS: Int\n    val STENCIL_BITS: Int\n    val POLYGON_OFFSET_UNITS: Int\n    val POLYGON_OFFSET_FACTOR: Int\n    val TEXTURE_BINDING_2D: Int\n    val SAMPLE_BUFFERS: Int\n    val SAMPLES: Int\n    val SAMPLE_COVERAGE_VALUE: Int\n    val SAMPLE_COVERAGE_INVERT: Int\n    val COMPRESSED_TEXTURE_FORMATS: Int\n    val DONT_CARE: Int\n    val FASTEST: Int\n    val NICEST: Int\n    val GENERATE_MIPMAP_HINT: Int\n    val BYTE: Int\n    val UNSIGNED_BYTE: Int\n    val SHORT: Int\n    val UNSIGNED_SHORT: Int\n    val INT: Int\n    val UNSIGNED_INT: Int\n    val FLOAT: Int\n    val DEPTH_COMPONENT: Int\n    val ALPHA: Int\n    val RGB: Int\n    val RGBA: Int\n    val LUMINANCE: Int\n    val LUMINANCE_ALPHA: Int\n    val UNSIGNED_SHORT_4_4_4_4: Int\n    val UNSIGNED_SHORT_5_5_5_1: Int\n    val UNSIGNED_SHORT_5_6_5: Int\n    val FRAGMENT_SHADER: Int\n    val VERTEX_SHADER: Int\n    val MAX_VERTEX_ATTRIBS: Int\n    val MAX_VERTEX_UNIFORM_VECTORS: Int\n    val MAX_VARYING_VECTORS: Int\n    val MAX_COMBINED_TEXTURE_IMAGE_UNITS: Int

```

MAX_VERTEX_TEXTURE_IMAGE_UNITS: Int\n val MAX_TEXTURE_IMAGE_UNITS: Int\n val
 MAX_FRAGMENT_UNIFORM_VECTORS: Int\n val SHADER_TYPE: Int\n val DELETE_STATUS:
 Int\n val LINK_STATUS: Int\n val VALIDATE_STATUS: Int\n val ATTACHED_SHADERS: Int\n
 val ACTIVE_UNIFORMS: Int\n val ACTIVE_ATTRIBUTES: Int\n val
 SHADING_LANGUAGE_VERSION: Int\n val CURRENT_PROGRAM: Int\n val NEVER: Int\n val
 LESS: Int\n val EQUAL: Int\n val LEQUAL: Int\n val GREATER: Int\n val NOTEQUAL: Int\n
 val GEQUAL: Int\n val ALWAYS: Int\n val KEEP: Int\n val REPLACE: Int\n val INCR: Int\n
 val DECR: Int\n val INVERT: Int\n val INCR_WRAP: Int\n val DECR_WRAP: Int\n val
 VENDOR: Int\n val RENDERER: Int\n val VERSION: Int\n val NEAREST: Int\n val LINEAR:
 Int\n val NEAREST_MIPMAP_NEAREST: Int\n val LINEAR_MIPMAP_NEAREST: Int\n val
 NEAREST_MIPMAP_LINEAR: Int\n val LINEAR_MIPMAP_LINEAR: Int\n val
 TEXTURE_MAG_FILTER: Int\n val TEXTURE_MIN_FILTER: Int\n val TEXTURE_WRAP_S: Int\n
 val TEXTURE_WRAP_T: Int\n val TEXTURE_2D: Int\n val TEXTURE: Int\n val
 TEXTURE_CUBE_MAP: Int\n val TEXTURE_BINDING_CUBE_MAP: Int\n val
 TEXTURE_CUBE_MAP_POSITIVE_X: Int\n val TEXTURE_CUBE_MAP_NEGATIVE_X: Int\n val
 TEXTURE_CUBE_MAP_POSITIVE_Y: Int\n val TEXTURE_CUBE_MAP_NEGATIVE_Y: Int\n val
 TEXTURE_CUBE_MAP_POSITIVE_Z: Int\n val TEXTURE_CUBE_MAP_NEGATIVE_Z: Int\n val
 MAX_CUBE_MAP_TEXTURE_SIZE: Int\n val TEXTURE0: Int\n val TEXTURE1: Int\n val
 TEXTURE2: Int\n val TEXTURE3: Int\n val TEXTURE4: Int\n val TEXTURE5: Int\n val
 TEXTURE6: Int\n val TEXTURE7: Int\n val TEXTURE8: Int\n val TEXTURE9: Int\n val
 TEXTURE10: Int\n val TEXTURE11: Int\n val TEXTURE12: Int\n val TEXTURE13: Int\n val
 TEXTURE14: Int\n val TEXTURE15: Int\n val TEXTURE16: Int\n val TEXTURE17: Int\n val
 TEXTURE18: Int\n val TEXTURE19: Int\n val TEXTURE20: Int\n val TEXTURE21: Int\n val
 TEXTURE22: Int\n val TEXTURE23: Int\n val TEXTURE24: Int\n val TEXTURE25: Int\n val
 TEXTURE26: Int\n val TEXTURE27: Int\n val TEXTURE28: Int\n val TEXTURE29: Int\n val
 TEXTURE30: Int\n val TEXTURE31: Int\n val ACTIVE_TEXTURE: Int\n val REPEAT: Int\n
 val CLAMP_TO_EDGE: Int\n val MIRRORED_REPEAT: Int\n val FLOAT_VEC2: Int\n val
 FLOAT_VEC3: Int\n val FLOAT_VEC4: Int\n val INT_VEC2: Int\n val INT_VEC3: Int\n val
 INT_VEC4: Int\n val BOOL: Int\n val BOOL_VEC2: Int\n val BOOL_VEC3: Int\n val
 BOOL_VEC4: Int\n val FLOAT_MAT2: Int\n val FLOAT_MAT3: Int\n val FLOAT_MAT4: Int\n
 val SAMPLER_2D: Int\n val SAMPLER_CUBE: Int\n val VERTEX_ATTRIB_ARRAY_ENABLED:
 Int\n val VERTEX_ATTRIB_ARRAY_SIZE: Int\n val VERTEX_ATTRIB_ARRAY_STRIDE: Int\n
 val VERTEX_ATTRIB_ARRAY_TYPE: Int\n val VERTEX_ATTRIB_ARRAY_NORMALIZED: Int\n
 val VERTEX_ATTRIB_ARRAY_POINTER: Int\n val VERTEX_ATTRIB_ARRAY_BUFFER_BINDING:
 Int\n val IMPLEMENTATION_COLOR_READ_TYPE: Int\n val
 IMPLEMENTATION_COLOR_READ_FORMAT: Int\n val COMPILE_STATUS: Int\n val
 LOW_FLOAT: Int\n val MEDIUM_FLOAT: Int\n val HIGH_FLOAT: Int\n val LOW_INT: Int\n
 val MEDIUM_INT: Int\n val HIGH_INT: Int\n val FRAMEBUFFER: Int\n val RENDERBUFFER:
 Int\n val RGBA4: Int\n val RGB5_A1: Int\n val RGB565: Int\n val DEPTH_COMPONENT16:
 Int\n val STENCIL_INDEX: Int\n val STENCIL_INDEX8: Int\n val DEPTH_STENCIL: Int\n val
 RENDERBUFFER_WIDTH: Int\n val RENDERBUFFER_HEIGHT: Int\n val
 RENDERBUFFER_INTERNAL_FORMAT: Int\n val RENDERBUFFER_RED_SIZE: Int\n val
 RENDERBUFFER_GREEN_SIZE: Int\n val RENDERBUFFER_BLUE_SIZE: Int\n val
 RENDERBUFFER_ALPHA_SIZE: Int\n val RENDERBUFFER_DEPTH_SIZE: Int\n val
 RENDERBUFFER_STENCIL_SIZE: Int\n val FRAMEBUFFER_ATTACHMENT_OBJECT_TYPE: Int\n
 val FRAMEBUFFER_ATTACHMENT_OBJECT_NAME: Int\n val
 FRAMEBUFFER_ATTACHMENT_TEXTURE_LEVEL: Int\n val
 FRAMEBUFFER_ATTACHMENT_TEXTURE_CUBE_MAP_FACE: Int\n val COLOR_ATTACHMENT0:

```

Int\n    val DEPTH_ATTACHMENT: Int\n    val STENCIL_ATTACHMENT: Int\n    val
DEPTH_STENCIL_ATTACHMENT: Int\n    val NONE: Int\n    val FRAMEBUFFER_COMPLETE: Int\n
val FRAMEBUFFER_INCOMPLETE_ATTACHMENT: Int\n    val
FRAMEBUFFER_INCOMPLETE_MISSING_ATTACHMENT: Int\n    val
FRAMEBUFFER_INCOMPLETE_DIMENSIONS: Int\n    val FRAMEBUFFER_UNSUPPORTED: Int\n
val FRAMEBUFFER_BINDING: Int\n    val RENDERBUFFER_BINDING: Int\n    val
MAX_RENDERBUFFER_SIZE: Int\n    val INVALID_FRAMEBUFFER_OPERATION: Int\n    val
UNPACK_FLIP_Y_WEBGL: Int\n    val UNPACK_PREMULTIPLY_ALPHA_WEBGL: Int\n    val
CONTEXT_LOST_WEBGL: Int\n    val UNPACK_COLORSPACE_CONVERSION_WEBGL: Int\n    val
BROWSER_DEFAULT_WEBGL: Int\n    })\n\n**\n * Exposes the JavaScript
[WebGLRenderingContext](https://developer.mozilla.org/en/docs/Web/API/WebGLRenderingContext) to Kotlin\n
*\npublic external abstract class WebGLRenderingContext : WebGLRenderingContextBase, RenderingContext {\n
companion object {\n    val DEPTH_BUFFER_BIT: Int\n    val STENCIL_BUFFER_BIT: Int\n    val
COLOR_BUFFER_BIT: Int\n    val POINTS: Int\n    val LINES: Int\n    val LINE_LOOP: Int\n    val
LINE_STRIP: Int\n    val TRIANGLES: Int\n    val TRIANGLE_STRIP: Int\n    val TRIANGLE_FAN:
Int\n    val ZERO: Int\n    val ONE: Int\n    val SRC_COLOR: Int\n    val ONE_MINUS_SRC_COLOR:
Int\n    val SRC_ALPHA: Int\n    val ONE_MINUS_SRC_ALPHA: Int\n    val DST_ALPHA: Int\n    val
ONE_MINUS_DST_ALPHA: Int\n    val DST_COLOR: Int\n    val ONE_MINUS_DST_COLOR: Int\n
val SRC_ALPHA_SATURATE: Int\n    val FUNC_ADD: Int\n    val BLEND_EQUATION: Int\n    val
BLEND_EQUATION_RGB: Int\n    val BLEND_EQUATION_ALPHA: Int\n    val FUNC_SUBTRACT:
Int\n    val FUNC_REVERSE_SUBTRACT: Int\n    val BLEND_DST_RGB: Int\n    val
BLEND_SRC_RGB: Int\n    val BLEND_DST_ALPHA: Int\n    val BLEND_SRC_ALPHA: Int\n    val
CONSTANT_COLOR: Int\n    val ONE_MINUS_CONSTANT_COLOR: Int\n    val CONSTANT_ALPHA:
Int\n    val ONE_MINUS_CONSTANT_ALPHA: Int\n    val BLEND_COLOR: Int\n    val
ARRAY_BUFFER: Int\n    val ELEMENT_ARRAY_BUFFER: Int\n    val ARRAY_BUFFER_BINDING:
Int\n    val ELEMENT_ARRAY_BUFFER_BINDING: Int\n    val STREAM_DRAW: Int\n    val
STATIC_DRAW: Int\n    val DYNAMIC_DRAW: Int\n    val BUFFER_SIZE: Int\n    val
BUFFER_USAGE: Int\n    val CURRENT_VERTEX_ATTRIB: Int\n    val FRONT: Int\n    val BACK:
Int\n    val FRONT_AND_BACK: Int\n    val CULL_FACE: Int\n    val BLEND: Int\n    val DITHER:
Int\n    val STENCIL_TEST: Int\n    val DEPTH_TEST: Int\n    val SCISSOR_TEST: Int\n    val
POLYGON_OFFSET_FILL: Int\n    val SAMPLE_ALPHA_TO_COVERAGE: Int\n    val
SAMPLE_COVERAGE: Int\n    val NO_ERROR: Int\n    val INVALID_ENUM: Int\n    val
INVALID_VALUE: Int\n    val INVALID_OPERATION: Int\n    val OUT_OF_MEMORY: Int\n    val CW:
Int\n    val CCW: Int\n    val LINE_WIDTH: Int\n    val ALIASED_POINT_SIZE_RANGE: Int\n    val
ALIASED_LINE_WIDTH_RANGE: Int\n    val CULL_FACE_MODE: Int\n    val FRONT_FACE: Int\n
val DEPTH_RANGE: Int\n    val DEPTH_WRITEMASK: Int\n    val DEPTH_CLEAR_VALUE: Int\n    val
DEPTH_FUNC: Int\n    val STENCIL_CLEAR_VALUE: Int\n    val STENCIL_FUNC: Int\n    val
STENCIL_FAIL: Int\n    val STENCIL_PASS_DEPTH_FAIL: Int\n    val STENCIL_PASS_DEPTH_PASS:
Int\n    val STENCIL_REF: Int\n    val STENCIL_VALUE_MASK: Int\n    val STENCIL_WRITEMASK:
Int\n    val STENCIL_BACK_FUNC: Int\n    val STENCIL_BACK_FAIL: Int\n    val
STENCIL_BACK_PASS_DEPTH_FAIL: Int\n    val STENCIL_BACK_PASS_DEPTH_PASS: Int\n    val
STENCIL_BACK_REF: Int\n    val STENCIL_BACK_VALUE_MASK: Int\n    val
STENCIL_BACK_WRITEMASK: Int\n    val VIEWPORT: Int\n    val SCISSOR_BOX: Int\n    val
COLOR_CLEAR_VALUE: Int\n    val COLOR_WRITEMASK: Int\n    val UNPACK_ALIGNMENT: Int\n
val PACK_ALIGNMENT: Int\n    val MAX_TEXTURE_SIZE: Int\n    val MAX_VIEWPORT_DIMS: Int\n
val SUBPIXEL_BITS: Int\n    val RED_BITS: Int\n    val GREEN_BITS: Int\n    val BLUE_BITS: Int\n
val ALPHA_BITS: Int\n    val DEPTH_BITS: Int\n    val STENCIL_BITS: Int\n    val
POLYGON_OFFSET_UNITS: Int\n    val POLYGON_OFFSET_FACTOR: Int\n    val

```

TEXTURE_BINDING_2D: Int\n val SAMPLE_BUFFERS: Int\n val SAMPLES: Int\n val
 SAMPLE_COVERAGE_VALUE: Int\n val SAMPLE_COVERAGE_INVERT: Int\n val
 COMPRESSED_TEXTURE_FORMATS: Int\n val DONT_CARE: Int\n val FASTEST: Int\n val
 NICEST: Int\n val GENERATE_MIPMAP_HINT: Int\n val BYTE: Int\n val UNSIGNED_BYTE:
 Int\n val SHORT: Int\n val UNSIGNED_SHORT: Int\n val INT: Int\n val UNSIGNED_INT: Int\n
 val FLOAT: Int\n val DEPTH_COMPONENT: Int\n val ALPHA: Int\n val RGB: Int\n val
 RGBA: Int\n val LUMINANCE: Int\n val LUMINANCE_ALPHA: Int\n val
 UNSIGNED_SHORT_4_4_4_4: Int\n val UNSIGNED_SHORT_5_5_5_1: Int\n val
 UNSIGNED_SHORT_5_6_5: Int\n val FRAGMENT_SHADER: Int\n val VERTEX_SHADER: Int\n
 val MAX_VERTEX_ATTRIBS: Int\n val MAX_VERTEX_UNIFORM_VECTORS: Int\n val
 MAX_VARYING_VECTORS: Int\n val MAX_COMBINED_TEXTURE_IMAGE_UNITS: Int\n val
 MAX_VERTEX_TEXTURE_IMAGE_UNITS: Int\n val MAX_TEXTURE_IMAGE_UNITS: Int\n val
 MAX_FRAGMENT_UNIFORM_VECTORS: Int\n val SHADER_TYPE: Int\n val DELETE_STATUS:
 Int\n val LINK_STATUS: Int\n val VALIDATE_STATUS: Int\n val ATTACHED_SHADERS: Int\n
 val ACTIVE_UNIFORMS: Int\n val ACTIVE_ATTRIBUTES: Int\n val
 SHADING_LANGUAGE_VERSION: Int\n val CURRENT_PROGRAM: Int\n val NEVER: Int\n val
 LESS: Int\n val EQUAL: Int\n val LEQUAL: Int\n val GREATER: Int\n val NOTEQUAL: Int\n
 val GEQUAL: Int\n val ALWAYS: Int\n val KEEP: Int\n val REPLACE: Int\n val INCR: Int\n
 val DECR: Int\n val INVERT: Int\n val INCR_WRAP: Int\n val DECR_WRAP: Int\n val
 VENDOR: Int\n val RENDERER: Int\n val VERSION: Int\n val NEAREST: Int\n val LINEAR:
 Int\n val NEAREST_MIPMAP_NEAREST: Int\n val LINEAR_MIPMAP_NEAREST: Int\n val
 NEAREST_MIPMAP_LINEAR: Int\n val LINEAR_MIPMAP_LINEAR: Int\n val
 TEXTURE_MAG_FILTER: Int\n val TEXTURE_MIN_FILTER: Int\n val TEXTURE_WRAP_S: Int\n
 val TEXTURE_WRAP_T: Int\n val TEXTURE_2D: Int\n val TEXTURE: Int\n val
 TEXTURE_CUBE_MAP: Int\n val TEXTURE_BINDING_CUBE_MAP: Int\n val
 TEXTURE_CUBE_MAP_POSITIVE_X: Int\n val TEXTURE_CUBE_MAP_NEGATIVE_X: Int\n val
 TEXTURE_CUBE_MAP_POSITIVE_Y: Int\n val TEXTURE_CUBE_MAP_NEGATIVE_Y: Int\n val
 TEXTURE_CUBE_MAP_POSITIVE_Z: Int\n val TEXTURE_CUBE_MAP_NEGATIVE_Z: Int\n val
 MAX_CUBE_MAP_TEXTURE_SIZE: Int\n val TEXTURE0: Int\n val TEXTURE1: Int\n val
 TEXTURE2: Int\n val TEXTURE3: Int\n val TEXTURE4: Int\n val TEXTURE5: Int\n val
 TEXTURE6: Int\n val TEXTURE7: Int\n val TEXTURE8: Int\n val TEXTURE9: Int\n val
 TEXTURE10: Int\n val TEXTURE11: Int\n val TEXTURE12: Int\n val TEXTURE13: Int\n val
 TEXTURE14: Int\n val TEXTURE15: Int\n val TEXTURE16: Int\n val TEXTURE17: Int\n val
 TEXTURE18: Int\n val TEXTURE19: Int\n val TEXTURE20: Int\n val TEXTURE21: Int\n val
 TEXTURE22: Int\n val TEXTURE23: Int\n val TEXTURE24: Int\n val TEXTURE25: Int\n val
 TEXTURE26: Int\n val TEXTURE27: Int\n val TEXTURE28: Int\n val TEXTURE29: Int\n val
 TEXTURE30: Int\n val TEXTURE31: Int\n val ACTIVE_TEXTURE: Int\n val REPEAT: Int\n
 val CLAMP_TO_EDGE: Int\n val MIRRORED_REPEAT: Int\n val FLOAT_VEC2: Int\n val
 FLOAT_VEC3: Int\n val FLOAT_VEC4: Int\n val INT_VEC2: Int\n val INT_VEC3: Int\n val
 INT_VEC4: Int\n val BOOL: Int\n val BOOL_VEC2: Int\n val BOOL_VEC3: Int\n val
 BOOL_VEC4: Int\n val FLOAT_MAT2: Int\n val FLOAT_MAT3: Int\n val FLOAT_MAT4: Int\n
 val SAMPLER_2D: Int\n val SAMPLER_CUBE: Int\n val VERTEX_ATTRIB_ARRAY_ENABLED:
 Int\n val VERTEX_ATTRIB_ARRAY_SIZE: Int\n val VERTEX_ATTRIB_ARRAY_STRIDE: Int\n
 val VERTEX_ATTRIB_ARRAY_TYPE: Int\n val VERTEX_ATTRIB_ARRAY_NORMALIZED: Int\n
 val VERTEX_ATTRIB_ARRAY_POINTER: Int\n val VERTEX_ATTRIB_ARRAY_BUFFER_BINDING:
 Int\n val IMPLEMENTATION_COLOR_READ_TYPE: Int\n val
 IMPLEMENTATION_COLOR_READ_FORMAT: Int\n val COMPILE_STATUS: Int\n val
 LOW_FLOAT: Int\n val MEDIUM_FLOAT: Int\n val HIGH_FLOAT: Int\n val LOW_INT: Int\n


```

\ "INVISIBLE_MEMBER" )\n@kotlin.internal.InlineOnly\npublic inline operator fun Int8Array.set(index: Int,
value: Byte) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Uint8Array](https://developer.mozilla.org/en/docs/Web/API/Uint8Array) to Kotlin\n *\npublic external open class
Uint8Array : ArrayBufferView {\n    constructor(length: Int)\n    constructor(array: Uint8Array)\n
constructor(array: Array<Byte>)\n    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length:
Int = definedExternally)\n    open val length: Int\n    override val buffer: ArrayBuffer\n    override val byteOffset:
Int\n    override val byteLength: Int\n    fun set(array: Uint8Array, offset: Int = definedExternally)\n    fun set(array:
Array<Byte>, offset: Int = definedExternally)\n    fun subarray(start: Int, end: Int): Uint8Array\n\n    companion
object {\n        val BYTES_PER_ELEMENT: Int\n    }\n}\n\n@Suppress(\ "INVISIBLE_REFERENCE" ),
\ "INVISIBLE_MEMBER" )\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint8Array.get(index: Int):
Byte = asDynamic()[index]\n\n@Suppress(\ "INVISIBLE_REFERENCE" ),
\ "INVISIBLE_MEMBER" )\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint8Array.set(index: Int,
value: Byte) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Uint8ClampedArray](https://developer.mozilla.org/en/docs/Web/API/Uint8ClampedArray) to Kotlin\n *\npublic
external open class Uint8ClampedArray : ArrayBufferView {\n    constructor(length: Int)\n    constructor(array:
Uint8ClampedArray)\n    constructor(array: Array<Byte>)\n    constructor(buffer: ArrayBuffer, byteOffset: Int =
definedExternally, length: Int = definedExternally)\n    open val length: Int\n    override val buffer: ArrayBuffer\n
override val byteOffset: Int\n    override val byteLength: Int\n    fun set(array: Uint8ClampedArray, offset: Int =
definedExternally)\n    fun set(array: Array<Byte>, offset: Int = definedExternally)\n    fun subarray(start: Int, end:
Int): Uint8ClampedArray\n\n    companion object {\n        val BYTES_PER_ELEMENT: Int\n    }\n}\n\n@Suppress(\ "INVISIBLE_REFERENCE" ),
\ "INVISIBLE_MEMBER" )\n@kotlin.internal.InlineOnly\npublic inline operator fun
Uint8ClampedArray.get(index: Int): Byte = asDynamic()[index]\n\n@Suppress(\ "INVISIBLE_REFERENCE" ),
\ "INVISIBLE_MEMBER" )\n@kotlin.internal.InlineOnly\npublic inline operator fun
Uint8ClampedArray.set(index: Int, value: Byte) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Int16Array](https://developer.mozilla.org/en/docs/Web/API/Int16Array) to Kotlin\n *\npublic external open class
Int16Array : ArrayBufferView {\n    constructor(length: Int)\n    constructor(array: Int16Array)\n
constructor(array: Array<Short>)\n    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length:
Int = definedExternally)\n    open val length: Int\n    override val buffer: ArrayBuffer\n    override val byteOffset:
Int\n    override val byteLength: Int\n    fun set(array: Int16Array, offset: Int = definedExternally)\n    fun set(array:
Array<Short>, offset: Int = definedExternally)\n    fun subarray(start: Int, end: Int): Int16Array\n\n    companion
object {\n        val BYTES_PER_ELEMENT: Int\n    }\n}\n\n@Suppress(\ "INVISIBLE_REFERENCE" ),
\ "INVISIBLE_MEMBER" )\n@kotlin.internal.InlineOnly\npublic inline operator fun Int16Array.get(index: Int):
Short = asDynamic()[index]\n\n@Suppress(\ "INVISIBLE_REFERENCE" ),
\ "INVISIBLE_MEMBER" )\n@kotlin.internal.InlineOnly\npublic inline operator fun Int16Array.set(index: Int,
value: Short) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Uint16Array](https://developer.mozilla.org/en/docs/Web/API/Uint16Array) to Kotlin\n *\npublic external open
class Uint16Array : ArrayBufferView {\n    constructor(length: Int)\n    constructor(array: Uint16Array)\n
constructor(array: Array<Short>)\n    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length:
Int = definedExternally)\n    open val length: Int\n    override val buffer: ArrayBuffer\n    override val byteOffset:
Int\n    override val byteLength: Int\n    fun set(array: Uint16Array, offset: Int = definedExternally)\n    fun set(array:
Array<Short>, offset: Int = definedExternally)\n    fun subarray(start: Int, end: Int): Uint16Array\n\n    companion
object {\n        val BYTES_PER_ELEMENT: Int\n    }\n}\n\n@Suppress(\ "INVISIBLE_REFERENCE" ),
\ "INVISIBLE_MEMBER" )\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint16Array.get(index: Int):
Short = asDynamic()[index]\n\n@Suppress(\ "INVISIBLE_REFERENCE" ),
\ "INVISIBLE_MEMBER" )\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint16Array.set(index: Int,
value: Short) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Int32Array](https://developer.mozilla.org/en/docs/Web/API/Int32Array) to Kotlin\n *\npublic external open class

```

```

Int32Array : ArrayBufferView {\n  constructor(length: Int)\n  constructor(array: Int32Array)\n
constructor(array: Array<Int>)\n  constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int
= definedExternally)\n  open val length: Int\n  override val buffer: ArrayBuffer\n  override val byteOffset: Int\n
override val byteLength: Int\n  fun set(array: Int32Array, offset: Int = definedExternally)\n  fun set(array:
Array<Int>, offset: Int = definedExternally)\n  fun subarray(start: Int, end: Int): Int32Array\n\n  companion object
{\n    val BYTES_PER_ELEMENT: Int\n  }\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int32Array.get(index: Int): Int
= asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int32Array.set(index: Int,
value: Int) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Uint32Array](https://developer.mozilla.org/en/docs/Web/API/Uint32Array) to Kotlin\n */\npublic external open
class Uint32Array : ArrayBufferView {\n  constructor(length: Int)\n  constructor(array: Uint32Array)\n
constructor(array: Array<Int>)\n  constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int
= definedExternally)\n  open val length: Int\n  override val buffer: ArrayBuffer\n  override val byteOffset: Int\n
override val byteLength: Int\n  fun set(array: Uint32Array, offset: Int = definedExternally)\n  fun set(array:
Array<Int>, offset: Int = definedExternally)\n  fun subarray(start: Int, end: Int): Uint32Array\n\n  companion
object {\n    val BYTES_PER_ELEMENT: Int\n  }\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint32Array.get(index: Int):
Int = asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint32Array.set(index: Int,
value: Int) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Float32Array](https://developer.mozilla.org/en/docs/Web/API/Float32Array) to Kotlin\n */\npublic external open
class Float32Array : ArrayBufferView {\n  constructor(length: Int)\n  constructor(array: Float32Array)\n
constructor(array: Array<Float>)\n  constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length:
Int = definedExternally)\n  open val length: Int\n  override val buffer: ArrayBuffer\n  override val byteOffset:
Int\n  override val byteLength: Int\n  fun set(array: Float32Array, offset: Int = definedExternally)\n  fun
set(array: Array<Float>, offset: Int = definedExternally)\n  fun subarray(start: Int, end: Int): Float32Array\n\n
companion object {\n    val BYTES_PER_ELEMENT: Int\n  }\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun Float32Array.get(index: Int):
Float = asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun Float32Array.set(index: Int,
value: Float) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Float64Array](https://developer.mozilla.org/en/docs/Web/API/Float64Array) to Kotlin\n */\npublic external open
class Float64Array : ArrayBufferView {\n  constructor(length: Int)\n  constructor(array: Float64Array)\n
constructor(array: Array<Double>)\n  constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length:
Int = definedExternally)\n  open val length: Int\n  override val buffer: ArrayBuffer\n  override val byteOffset:
Int\n  override val byteLength: Int\n  fun set(array: Float64Array, offset: Int = definedExternally)\n  fun
set(array: Array<Double>, offset: Int = definedExternally)\n  fun subarray(start: Int, end: Int): Float64Array\n\n
companion object {\n    val BYTES_PER_ELEMENT: Int\n  }\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun Float64Array.get(index: Int):
Double = asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun Float64Array.set(index: Int,
value: Double) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[DataView](https://developer.mozilla.org/en/docs/Web/API/DataView) to Kotlin\n */\npublic external open class
DataView(buffer: ArrayBuffer, byteOffset: Int = definedExternally, byteLength: Int = definedExternally) :
ArrayBufferView {\n  override val buffer: ArrayBuffer\n  override val byteOffset: Int\n  override val

```

```

byteLength: Int\n fun getInt8(byteOffset: Int): Byte\n fun getUInt8(byteOffset: Int): Byte\n fun
getInt16(byteOffset: Int, littleEndian: Boolean = definedExternally): Short\n fun getUInt16(byteOffset: Int,
littleEndian: Boolean = definedExternally): Short\n fun getInt32(byteOffset: Int, littleEndian: Boolean =
definedExternally): Int\n fun getUInt32(byteOffset: Int, littleEndian: Boolean = definedExternally): Int\n fun
getFloat32(byteOffset: Int, littleEndian: Boolean = definedExternally): Float\n fun getFloat64(byteOffset: Int,
littleEndian: Boolean = definedExternally): Double\n fun setInt8(byteOffset: Int, value: Byte)\n fun
setUInt8(byteOffset: Int, value: Byte)\n fun setInt16(byteOffset: Int, value: Short, littleEndian: Boolean =
definedExternally)\n fun setUInt16(byteOffset: Int, value: Short, littleEndian: Boolean = definedExternally)\n
fun setInt32(byteOffset: Int, value: Int, littleEndian: Boolean = definedExternally)\n fun setUInt32(byteOffset: Int,
value: Int, littleEndian: Boolean = definedExternally)\n fun setFloat32(byteOffset: Int, value: Float, littleEndian:
Boolean = definedExternally)\n fun setFloat64(byteOffset: Int, value: Double, littleEndian: Boolean =
definedExternally)\n\n\npublic external interface BufferDataSource\n\n\npublic external interface
TexImageSource", "/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n *
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for
details\n\npackage org.w3c.dom.clipboard\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport
org.w3c.dom.*\nimport org.w3c.dom.events.*\n\npublic external interface ClipboardEventInit : EventInit {\n var
clipboardData: DataTransfer? /* = null */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun ClipboardEventInit(clipboardData:
DataTransfer? = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false):
ClipboardEventInit {\n val o = js("{}")\n o["clipboardData"] = clipboardData\n o["bubbles"] = bubbles\n
o["cancelable"] = cancelable\n o["composed"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[ClipboardEvent](https://developer.mozilla.org/en/docs/Web/API/ClipboardEvent) to Kotlin\n */\n\npublic external
open class ClipboardEvent(type: String, eventInitDict: ClipboardEventInit = definedExternally) : Event {\n open
val clipboardData: DataTransfer?\n\n companion object {\n val NONE: Short\n val
CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n
}\n}\n\n/**\n * Exposes the JavaScript [Clipboard](https://developer.mozilla.org/en/docs/Web/API/Clipboard) to
Kotlin\n */\n\npublic external abstract class Clipboard : EventTarget {\n fun read(): Promise<DataTransfer>\n fun
readText(): Promise<String>\n fun write(data: DataTransfer): Promise<Unit>\n fun writeText(data: String):
Promise<Unit>\n}\n\npublic external interface ClipboardPermissionDescriptor {\n var allowWithoutGesture:
Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun
ClipboardPermissionDescriptor(allowWithoutGesture: Boolean? = false): ClipboardPermissionDescriptor {\n val
o = js("{}")\n o["allowWithoutGesture"] = allowWithoutGesture\n return o\n}, "/*\n * Copyright 2010-
2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-
GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.dom.css\n\nimport
kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\n\npublic external abstract class MediaList :
ItemArrayLike<String> {\n open var mediaText: String\n fun appendMedium(medium: String)\n fun
deleteMedium(medium: String)\n override fun item(index: Int):
String?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun MediaList.get(index: Int):
String? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[StyleSheet](https://developer.mozilla.org/en/docs/Web/API/StyleSheet) to Kotlin\n */\n\npublic external abstract
class StyleSheet {\n open val type: String\n open val href: String?\n open val ownerNode:
UnionElementOrProcessingInstruction?\n open val parentStyleSheet: StyleSheet?\n open val title: String?\n

```



```

open val media: MediaList\n  open var disabled: Boolean\n}\n\n/**\n * Exposes the JavaScript
[CSSStyleSheet](https://developer.mozilla.org/en/docs/Web/API/CSSStyleSheet) to Kotlin\n */\npublic external
abstract class CSSStyleSheet : StyleSheet {\n  open val ownerRule: CSSRule?\n  open val cssRules:
CSSRuleList\n  fun insertRule(rule: String, index: Int): Int\n  fun deleteRule(index: Int)\n}\n\n/**\n * Exposes the
JavaScript [StyleSheetList](https://developer.mozilla.org/en/docs/Web/API/StyleSheetList) to Kotlin\n */\npublic
external abstract class StyleSheetList : ItemArrayLike<StyleSheet> {\n  override fun item(index: Int):
StyleSheet?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun StyleSheetList.get(index: Int):
StyleSheet? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[LinkStyle](https://developer.mozilla.org/en/docs/Web/API/LinkStyle) to Kotlin\n */\npublic external interface
LinkStyle {\n  val sheet: StyleSheet?\n  get() = definedExternally\n}\n\n/**\n * Exposes the JavaScript
[CSSRuleList](https://developer.mozilla.org/en/docs/Web/API/CSSRuleList) to Kotlin\n */\npublic external abstract
class CSSRuleList : ItemArrayLike<CSSRule> {\n  override fun item(index: Int):
CSSRule?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun CSSRuleList.get(index: Int):
CSSRule? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[CSSRule](https://developer.mozilla.org/en/docs/Web/API/CSSRule) to Kotlin\n */\npublic external abstract class
CSSRule {\n  open val type: Short\n  open var cssText: String\n  open val parentRule: CSSRule?\n  open val
parentStyleSheet: CSSStyleSheet?\n\n  companion object {\n    val STYLE_RULE: Short\n    val
CHARSET_RULE: Short\n    val IMPORT_RULE: Short\n    val MEDIA_RULE: Short\n    val
FONT_FACE_RULE: Short\n    val PAGE_RULE: Short\n    val MARGIN_RULE: Short\n    val
NAMESPACE_RULE: Short\n  }\n}\n\n/**\n * Exposes the JavaScript
[CSSStyleRule](https://developer.mozilla.org/en/docs/Web/API/CSSStyleRule) to Kotlin\n */\npublic external
abstract class CSSStyleRule : CSSRule {\n  open var selectorText: String\n  open val style:
CSSStyleDeclaration\n\n  companion object {\n    val STYLE_RULE: Short\n    val CHARSET_RULE:
Short\n    val IMPORT_RULE: Short\n    val MEDIA_RULE: Short\n    val FONT_FACE_RULE: Short\n
val PAGE_RULE: Short\n    val MARGIN_RULE: Short\n    val NAMESPACE_RULE: Short\n  }\n}\n\npublic external abstract class CSSImportRule : CSSRule {\n  open val href: String\n  open val media:
MediaList\n  open val styleSheet: CSSStyleSheet\n\n  companion object {\n    val STYLE_RULE: Short\n
val CHARSET_RULE: Short\n    val IMPORT_RULE: Short\n    val MEDIA_RULE: Short\n    val
FONT_FACE_RULE: Short\n    val PAGE_RULE: Short\n    val MARGIN_RULE: Short\n    val
NAMESPACE_RULE: Short\n  }\n}\n\n/**\n * Exposes the JavaScript
[CSSGroupingRule](https://developer.mozilla.org/en/docs/Web/API/CSSGroupingRule) to Kotlin\n */\npublic
external abstract class CSSGroupingRule : CSSRule {\n  open val cssRules: CSSRuleList\n  fun insertRule(rule:
String, index: Int): Int\n  fun deleteRule(index: Int)\n\n  companion object {\n    val STYLE_RULE: Short\n
val CHARSET_RULE: Short\n    val IMPORT_RULE: Short\n    val MEDIA_RULE: Short\n    val
FONT_FACE_RULE: Short\n    val PAGE_RULE: Short\n    val MARGIN_RULE: Short\n    val
NAMESPACE_RULE: Short\n  }\n}\n\n/**\n * Exposes the JavaScript
[CSSMediaRule](https://developer.mozilla.org/en/docs/Web/API/CSSMediaRule) to Kotlin\n */\npublic external
abstract class CSSMediaRule : CSSGroupingRule {\n  open val media: MediaList\n\n  companion object {\n
val STYLE_RULE: Short\n    val CHARSET_RULE: Short\n    val IMPORT_RULE: Short\n    val
MEDIA_RULE: Short\n    val FONT_FACE_RULE: Short\n    val PAGE_RULE: Short\n    val
MARGIN_RULE: Short\n    val NAMESPACE_RULE: Short\n  }\n}\n\n/**\n * Exposes the JavaScript
[CSSPageRule](https://developer.mozilla.org/en/docs/Web/API/CSSPageRule) to Kotlin\n */\npublic external
abstract class CSSPageRule : CSSGroupingRule {\n  open var selectorText: String\n  open val style:
CSSStyleDeclaration\n\n  companion object {\n    val STYLE_RULE: Short\n    val CHARSET_RULE:
Short\n    val IMPORT_RULE: Short\n    val MEDIA_RULE: Short\n    val FONT_FACE_RULE: Short\n
val PAGE_RULE: Short\n    val MARGIN_RULE: Short\n    val NAMESPACE_RULE: Short\n  }\n}

```

```
}  
}\n\npublic external abstract class CSSMarginRule : CSSRule {\n  open val name: String\n  open val style: CSSStyleDeclaration\n\n  companion object {\n    val STYLE_RULE: Short\n    val CHARSET_RULE: Short\n    val IMPORT_RULE: Short\n    val MEDIA_RULE: Short\n    val FONT_FACE_RULE: Short\n    val PAGE_RULE: Short\n    val MARGIN_RULE: Short\n    val NAMESPACE_RULE: Short\n  }\n}\n\n/**\n * Exposes the JavaScript
```

```
[CSSNamespaceRule](https://developer.mozilla.org/en/docs/Web/API/CSSNamespaceRule) to Kotlin\n */\n\npublic external abstract class CSSNamespaceRule : CSSRule {\n  open val namespaceURI: String\n  open val prefix: String\n\n  companion object {\n    val STYLE_RULE: Short\n    val CHARSET_RULE: Short\n    val IMPORT_RULE: Short\n    val MEDIA_RULE: Short\n    val FONT_FACE_RULE: Short\n    val PAGE_RULE: Short\n    val MARGIN_RULE: Short\n    val NAMESPACE_RULE: Short\n  }\n}\n\n/**\n * Exposes the JavaScript
```

```
[CSSStyleDeclaration](https://developer.mozilla.org/en/docs/Web/API/CSSStyleDeclaration) to Kotlin\n */\n\npublic external abstract class CSSStyleDeclaration : ItemArrayLike<String> {\n  open var cssText: String\n  open val parentRule: CSSRule?\n  open var cssFloat: String\n  open var alignContent: String\n  open var alignItems: String\n  open var alignSelf: String\n  open var animation: String\n  open var animationDelay: String\n  open var animationDirection: String\n  open var animationDuration: String\n  open var animationFillMode: String\n  open var animationIterationCount: String\n  open var animationName: String\n  open var animationPlayState: String\n  open var animationTimingFunction: String\n  open var backfaceVisibility: String\n  open var background: String\n  open var backgroundAttachment: String\n  open var backgroundClip: String\n  open var backgroundColor: String\n  open var backgroundImage: String\n  open var backgroundOrigin: String\n  open var backgroundPosition: String\n  open var backgroundRepeat: String\n  open var backgroundSize: String\n  open var border: String\n  open var borderBottom: String\n  open var borderBottomColor: String\n  open var borderBottomLeftRadius: String\n  open var borderBottomRightRadius: String\n  open var borderBottomStyle: String\n  open var borderBottomWidth: String\n  open var borderCollapse: String\n  open var borderColor: String\n  open var borderImage: String\n  open var borderImageOutset: String\n  open var borderImageRepeat: String\n  open var borderImageSlice: String\n  open var borderImageSource: String\n  open var borderImageWidth: String\n  open var borderLeft: String\n  open var borderLeftColor: String\n  open var borderLeftStyle: String\n  open var borderLeftWidth: String\n  open var borderRadius: String\n  open var borderRight: String\n  open var borderRightColor: String\n  open var borderRightStyle: String\n  open var borderRightWidth: String\n  open var borderSpacing: String\n  open var borderStyle: String\n  open var borderTop: String\n  open var borderTopColor: String\n  open var borderTopLeftRadius: String\n  open var borderTopRightRadius: String\n  open var borderTopStyle: String\n  open var borderTopWidth: String\n  open var borderWidth: String\n  open var bottom: String\n  open var boxDecorationBreak: String\n  open var boxShadow: String\n  open var boxSizing: String\n  open var breakAfter: String\n  open var breakBefore: String\n  open var breakInside: String\n  open var captionSide: String\n  open var clear: String\n  open var clip: String\n  open var color: String\n  open var columnCount: String\n  open var columnFill: String\n  open var columnGap: String\n  open var columnRule: String\n  open var columnRuleColor: String\n  open var columnRuleStyle: String\n  open var columnRuleWidth: String\n  open var columnSpan: String\n  open var columnWidth: String\n  open var columns: String\n  open var content: String\n  open var counterIncrement: String\n  open var counterReset: String\n  open var cursor: String\n  open var direction: String\n  open var display: String\n  open var emptyCells: String\n  open var filter: String\n  open var flex: String\n  open var flexBasis: String\n  open var flexDirection: String\n  open var flexFlow: String\n  open var flexGrow: String\n  open var flexShrink: String\n  open var flexWrap: String\n  open var font: String\n  open var fontFamily: String\n  open var fontFeatureSettings: String\n  open var fontKerning: String\n  open var fontLanguageOverride: String\n  open var fontSize: String\n  open var fontSizeAdjust: String\n  open var fontStretch: String\n  open var fontStyle: String\n  open var fontSynthesis: String\n  open var fontVariant: String\n  open var fontVariantAlternates: String\n  open var fontVariantCaps: String\n  open var fontVariantEastAsian: String\n  open var fontVariantLigatures: String\n  open var fontVariantNumeric: String
```

```

open var fontVariantPosition: String\n open var fontWeight: String\n open var hangingPunctuation: String\n
open var height: String\n open var hyphens: String\n open var imageOrientation: String\n open var
imageRendering: String\n open var imageResolution: String\n open var imeMode: String\n open var
justifyContent: String\n open var left: String\n open var letterSpacing: String\n open var lineBreak: String\n
open var lineHeight: String\n open var listStyle: String\n open var listStyleImage: String\n open var
listStylePosition: String\n open var listStyleType: String\n open var margin: String\n open var marginBottom:
String\n open var marginLeft: String\n open var marginRight: String\n open var marginTop: String\n open
var mark: String\n open var markAfter: String\n open var markBefore: String\n open var marks: String\n
open var marqueeDirection: String\n open var marqueePlayCount: String\n open var marqueeSpeed: String\n
open var marqueeStyle: String\n open var mask: String\n open var maskType: String\n open var maxHeight:
String\n open var maxWidth: String\n open var minHeight: String\n open var minWidth: String\n open var
navDown: String\n open var navIndex: String\n open var navLeft: String\n open var navRight: String\n open
var navUp: String\n open var objectFit: String\n open var objectPosition: String\n open var opacity: String\n
open var order: String\n open var orphans: String\n open var outline: String\n open var outlineColor: String\n
open var outlineOffset: String\n open var outlineStyle: String\n open var outlineWidth: String\n open var
overflowWrap: String\n open var overflowX: String\n open var overflowY: String\n open var padding:
String\n open var paddingBottom: String\n open var paddingLeft: String\n open var paddingRight: String\n
open var paddingTop: String\n open var pageBreakAfter: String\n open var pageBreakBefore: String\n open
var pageBreakInside: String\n open var perspective: String\n open var perspectiveOrigin: String\n open var
phonemes: String\n open var position: String\n open var quotes: String\n open var resize: String\n open var
rest: String\n open var restAfter: String\n open var restBefore: String\n open var right: String\n open var
tabSize: String\n open var tableLayout: String\n open var textAlign: String\n open var textAlignLast: String\n
open var textCombineUpright: String\n open var textDecoration: String\n open var textDecorationColor:
String\n open var textDecorationLine: String\n open var textDecorationStyle: String\n open var textIndent:
String\n open var textJustify: String\n open var textOrientation: String\n open var textOverflow: String\n
open var textShadow: String\n open var textTransform: String\n open var textUnderlinePosition: String\n open
var top: String\n open var transform: String\n open var transformOrigin: String\n open var transformStyle:
String\n open var transition: String\n open var transitionDelay: String\n open var transitionDuration: String\n
open var transitionProperty: String\n open var transitionTimingFunction: String\n open var unicodeBidi:
String\n open var verticalAlign: String\n open var visibility: String\n open var voiceBalance: String\n open
var voiceDuration: String\n open var voicePitch: String\n open var voicePitchRange: String\n open var
voiceRate: String\n open var voiceStress: String\n open var voiceVolume: String\n open var whiteSpace:
String\n open var widows: String\n open var width: String\n open var wordBreak: String\n open var
wordSpacing: String\n open var wordWrap: String\n open var writingMode: String\n open var zIndex: String\n
open var _dashed_attribute: String\n open var _camel_cased_attribute: String\n open var
_webkit_cased_attribute: String\n fun getPropertyValue(property: String): String\n fun
getPropertyPriority(property: String): String\n fun setProperty(property: String, value: String, priority: String =
definedExternally)\n fun setPropertyValue(property: String, value: String)\n fun setPropertyPriority(property:
String, priority: String)\n fun removeProperty(property: String): String\n override fun item(index: Int):
String\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun
CSSStyleDeclaration.get(index: Int): String? = asDynamic()[index]\n\npublic external interface
ElementCSSInlineStyle {\n val style: CSSStyleDeclaration\n}\n\n/**\n * Exposes the JavaScript
[CSS](https://developer.mozilla.org/en/docs/Web/API/CSS) to Kotlin\n */\npublic external abstract class CSS {\n
companion object {\n fun escape(ident: String): String\n }\n}\n\npublic external interface
UnionElementOrProcessingInstruction, "/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See

```

```

github.com/kotlin/dukat for details\n\npackage org.w3c.dom.encryptedmedia\n\nimport kotlin.js.*\nimport
org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\n\n/**\n * Exposes the JavaScript
[MediaKeySystemConfiguration](https://developer.mozilla.org/en/docs/Web/API/MediaKeySystemConfiguration)
to Kotlin\n\npublic external interface MediaKeySystemConfiguration {\n    var label: String? /* = "" */\n    get() = definedExternally\n    set(value) = definedExternally\n    var initDataTypes: Array<String>? /* = arrayOf()
*/\n    get() = definedExternally\n    set(value) = definedExternally\n    var audioCapabilities:
Array<MediaKeySystemMediaCapability>? /* = arrayOf() */\n    get() = definedExternally\n    set(value) =
definedExternally\n    var videoCapabilities: Array<MediaKeySystemMediaCapability>? /* = arrayOf() */\n    get() = definedExternally\n    set(value) = definedExternally\n    var distinctiveIdentifier:
MediaKeysRequirement? /* = MediaKeysRequirement.OPTIONAL */\n    get() = definedExternally\n    set(value) = definedExternally\n    var persistentState: MediaKeysRequirement? /* =
MediaKeysRequirement.OPTIONAL */\n    get() = definedExternally\n    set(value) = definedExternally\n    var sessionTypes: Array<String>?\n    get() = definedExternally\n    set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun MediaKeySystemConfiguration(label:
String? = "", initDataTypes: Array<String>? = arrayOf(), audioCapabilities:
Array<MediaKeySystemMediaCapability>? = arrayOf(), videoCapabilities:
Array<MediaKeySystemMediaCapability>? = arrayOf(), distinctiveIdentifier: MediaKeysRequirement? =
MediaKeysRequirement.OPTIONAL, persistentState: MediaKeysRequirement? =
MediaKeysRequirement.OPTIONAL, sessionTypes: Array<String>? = undefined): MediaKeySystemConfiguration
{\n    val o = js("{}")\n    o["label"] = label\n    o["initDataTypes"] = initDataTypes\n
o["audioCapabilities"] = audioCapabilities\n    o["videoCapabilities"] = videoCapabilities\n
o["distinctiveIdentifier"] = distinctiveIdentifier\n    o["persistentState"] = persistentState\n    o["sessionTypes"]
= sessionTypes\n    return o\n}\n\npublic external interface MediaKeySystemMediaCapability {\n    var
contentType: String? /* = "" */\n    get() = definedExternally\n    set(value) = definedExternally\n    var
robustness: String? /* = "" */\n    get() = definedExternally\n    set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun
MediaKeySystemMediaCapability(contentType: String? = "", robustness: String? = ""):
MediaKeySystemMediaCapability {\n    val o = js("{}")\n    o["contentType"] = contentType\n
o["robustness"] = robustness\n    return o\n}\n\n/**\n * Exposes the JavaScript
[MediaKeySystemAccess](https://developer.mozilla.org/en/docs/Web/API/MediaKeySystemAccess) to Kotlin\n\n
public external abstract class MediaKeySystemAccess {\n    open val keySystem: String\n    fun
getConfiguration(): MediaKeySystemConfiguration\n    fun createMediaKeys(): Promise<MediaKeys>\n}\n\n/**\n * Exposes the JavaScript [MediaKeys](https://developer.mozilla.org/en/docs/Web/API/MediaKeys) to Kotlin\n\n
public external abstract class MediaKeys {\n    fun createSession(sessionType: MediaKeySessionType =
definedExternally): MediaKeySession\n    fun setServerCertificate(serverCertificate: dynamic):
Promise<Boolean>\n}\n\n/**\n * Exposes the JavaScript
[MediaKeySession](https://developer.mozilla.org/en/docs/Web/API/MediaKeySession) to Kotlin\n\n
public external abstract class MediaKeySession : EventTarget {\n    open val sessionId: String\n    open val expiration:
Double\n    open val closed: Promise<Unit>\n    open val keyStatuses: MediaKeyStatusMap\n    open var
onkeystatuschange: ((Event) -> dynamic)?\n    open var onmessage: ((MessageEvent) -> dynamic)?\n    fun
generateRequest(initDataType: String, initData: dynamic): Promise<Unit>\n    fun load(sessionId: String):
Promise<Boolean>\n    fun update(response: dynamic): Promise<Unit>\n    fun close(): Promise<Unit>\n    fun
remove(): Promise<Unit>\n}\n\n/**\n * Exposes the JavaScript
[MediaKeyStatusMap](https://developer.mozilla.org/en/docs/Web/API/MediaKeyStatusMap) to Kotlin\n\n
public external abstract class MediaKeyStatusMap {\n    open val size: Int\n    fun has(keyId: dynamic): Boolean\n    fun
get(keyId: dynamic): Any?\n}\n\n/**\n * Exposes the JavaScript

```

```

[MediaKeyMessageEvent](https://developer.mozilla.org/en/docs/Web/API/MediaKeyMessageEvent) to Kotlin\n
*\npublic external open class MediaKeyMessageEvent(type: String, eventInitDict: MediaKeyMessageEventInit) :
Event {\n  open val messageType: MediaKeyMessageType\n  open val message: ArrayBuffer\n\n  companion
object {\n    val NONE: Short\n    val CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val
BUBBLING_PHASE: Short\n  }\n}\n\npublic external interface MediaKeyMessageEventInit : EventInit {\n  var
messageType: MediaKeyMessageType?\n  var message:
ArrayBuffer?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun
MediaKeyMessageEventInit(messageType: MediaKeyMessageType?, message: ArrayBuffer?, bubbles: Boolean? =
false, cancelable: Boolean? = false, composed: Boolean? = false): MediaKeyMessageEventInit {\n  val o =
js(\"({})\")\n  o[\"messageType\"] = messageType\n  o[\"message\"] = message\n  o[\"bubbles\"] = bubbles\n
o[\"cancelable\"] = cancelable\n  o[\"composed\"] = composed\n  return o\n}\n\npublic external open class
MediaEncryptedEvent(type: String, eventInitDict: MediaEncryptedEventInit = definedExternally) : Event {\n
open val initDataType: String\n  open val initData: ArrayBuffer?\n\n  companion object {\n    val NONE:
Short\n    val CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val BUBBLING_PHASE:
Short\n  }\n}\n\npublic external interface MediaEncryptedEventInit : EventInit {\n  var initDataType: String? /* =
\"\" *\n  get() = definedExternally\n  set(value) = definedExternally\n  var initData: ArrayBuffer? /* = null
*\n  get() = definedExternally\n  set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun
MediaEncryptedEventInit(initDataType: String? = \"\", initData: ArrayBuffer? = null, bubbles: Boolean? = false,
cancelable: Boolean? = false, composed: Boolean? = false): MediaEncryptedEventInit {\n  val o = js(\"({})\")\n
o[\"initDataType\"] = initData\n  o[\"initData\"] = initData\n  o[\"bubbles\"] = bubbles\n  o[\"cancelable\"]
= cancelable\n  o[\"composed\"] = composed\n  return o\n}\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface MediaKeysRequirement {\n  companion object\n}\n\npublic inline val
MediaKeysRequirement.Companion.REQUIRED: MediaKeysRequirement get() =
\"required\".asDynamic().unsafeCast<MediaKeysRequirement>()\n\npublic inline val
MediaKeysRequirement.Companion.OPTIONAL: MediaKeysRequirement get() =
\"optional\".asDynamic().unsafeCast<MediaKeysRequirement>()\n\npublic inline val
MediaKeysRequirement.Companion.NOT_ALLOWED: MediaKeysRequirement get() = \"not-
allowed\".asDynamic().unsafeCast<MediaKeysRequirement>()\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface MediaKeySessionType {\n  companion object\n}\n\npublic inline val
MediaKeySessionType.Companion.TEMPORARY: MediaKeySessionType get() =
\"temporary\".asDynamic().unsafeCast<MediaKeySessionType>()\n\npublic inline val
MediaKeySessionType.Companion.PERSISTENT_LICENSE: MediaKeySessionType get() = \"persistent-
license\".asDynamic().unsafeCast<MediaKeySessionType>()\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface MediaKeyStatus {\n  companion object\n}\n\npublic inline val MediaKeyStatus.Companion.USABLE:
MediaKeyStatus get() = \"usable\".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.EXPIRED: MediaKeyStatus get() =
\"expired\".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.RELEASED: MediaKeyStatus get() =
\"released\".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.OUTPUT_RESTRICTED: MediaKeyStatus get() = \"output-
restricted\".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.OUTPUT_DOWNSCALED: MediaKeyStatus get() = \"output-

```

```

downscaled\`.asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.STATUS_PENDING: MediaKeyStatus get() = `status-
pending`\`.asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.INTERNAL_ERROR: MediaKeyStatus get() = `internal-
error`\`.asDynamic().unsafeCast<MediaKeyStatus>()\n\n/* please, don't implement this interface!
*\n@JsName(`null`)\n@Suppress(`NESTED_CLASS_IN_EXTERNAL_INTERFACE`)\n\npublic external
interface MediaKeyMessageType {\n    companion object\n}\n\npublic inline val
MediaKeyMessageType.Companion.LICENSE_REQUEST: MediaKeyMessageType get() = `license-
request`\`.asDynamic().unsafeCast<MediaKeyMessageType>()\n\npublic inline val
MediaKeyMessageType.Companion.LICENSE_RENEWAL: MediaKeyMessageType get() = `license-
renewal`\`.asDynamic().unsafeCast<MediaKeyMessageType>()\n\npublic inline val
MediaKeyMessageType.Companion.LICENSE_RELEASE: MediaKeyMessageType get() = `license-
release`\`.asDynamic().unsafeCast<MediaKeyMessageType>()\n\npublic inline val
MediaKeyMessageType.Companion.INDIVIDUALIZATION_REQUEST: MediaKeyMessageType get() =
`individualization-request`\`.asDynamic().unsafeCast<MediaKeyMessageType>()`/*\n * Copyright 2010-2021
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n\n// NOTE: THIS FILE IS AUTO-
GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage
org.w3c.dom.events\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\n\n/**\n * Exposes
the JavaScript [UIEvent](https://developer.mozilla.org/en/docs/Web/API/UIEvent) to Kotlin\n *\n\npublic external
open class UIEvent(type: String, eventInitDict: UIEventInit = definedExternally) : Event {\n    open val view:
Window?\n    open val detail: Int?\n    companion object {\n        val NONE: Short\n        val
CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n
    }\n}\n\npublic external interface UIEventInit : EventInit {\n    var view: Window? /* = null *\n    get() =
definedExternally\n    set(value) = definedExternally\n    var detail: Int? /* = 0 *\n    get() =
definedExternally\n    set(value) = definedExternally\n}\n\n@Suppress(`INVISIBLE_REFERENCE`,
`INVISIBLE_MEMBER`)\n\n@kotlin.internal.InlineOnly\n\npublic inline fun UIEventInit(view: Window? = null,
detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): UIEventInit
{\n    val o = js(`{}`)\n    o["view"] = view\n    o["detail"] = detail\n    o["bubbles"] = bubbles\n
o["cancelable"] = cancelable\n    o["composed"] = composed\n    return o\n}\n\n/**\n * Exposes the JavaScript
[FocusEvent](https://developer.mozilla.org/en/docs/Web/API/FocusEvent) to Kotlin\n *\n\npublic external open class
FocusEvent(type: String, eventInitDict: FocusEventInit = definedExternally) : UIEvent {\n    open val relatedTarget:
EventTarget?\n\n    companion object {\n        val NONE: Short\n        val CAPTURING_PHASE: Short\n        val
AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }\n}\n\npublic external interface FocusEventInit :
UIEventInit {\n    var relatedTarget: EventTarget? /* = null *\n    get() = definedExternally\n    set(value) =
definedExternally\n}\n\n@Suppress(`INVISIBLE_REFERENCE`,
`INVISIBLE_MEMBER`)\n\n@kotlin.internal.InlineOnly\n\npublic inline fun FocusEventInit(relatedTarget:
EventTarget? = null, view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? =
false, composed: Boolean? = false): FocusEventInit {\n    val o = js(`{}`)\n    o["relatedTarget"] =
relatedTarget\n    o["view"] = view\n    o["detail"] = detail\n    o["bubbles"] = bubbles\n    o["cancelable"] =
cancelable\n    o["composed"] = composed\n    return o\n}\n\n/**\n * Exposes the JavaScript
[MouseEvent](https://developer.mozilla.org/en/docs/Web/API/MouseEvent) to Kotlin\n *\n\npublic external open
class MouseEvent(type: String, eventInitDict: MouseEventInit = definedExternally) : UIEvent,
UnionElementOrMouseEvent {\n    open val screenX: Int\n    open val screenY: Int\n    open val clientX: Int\n
open val clientY: Int\n    open val ctrlKey: Boolean\n    open val shiftKey: Boolean\n    open val altKey: Boolean\n
open val metaKey: Boolean\n    open val button: Short\n    open val buttons: Short\n    open val relatedTarget:
EventTarget?\n    open val region: String?\n    open val pageX: Double\n    open val pageY: Double\n    open val x:
Double\n    open val y: Double\n    open val offsetX: Double\n    open val offsetY: Double\n    fun

```

```

getModifierState(keyArg: String): Boolean\n\n companion object {\n    val NONE: Short\n    val CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val BUBBLING_PHASE: Short\n}\n\npublic external interface MouseEventInit : EventModifierInit {\n    var screenX: Int? /* = 0 */\n    get() = definedExternally\n    set(value) = definedExternally\n    var screenY: Int? /* = 0 */\n    get() = definedExternally\n    set(value) = definedExternally\n    var clientX: Int? /* = 0 */\n    get() = definedExternally\n    set(value) = definedExternally\n    var clientY: Int? /* = 0 */\n    get() = definedExternally\n    set(value) = definedExternally\n    var button: Short? /* = 0 */\n    get() = definedExternally\n    set(value) = definedExternally\n    var buttons: Short? /* = 0 */\n    get() = definedExternally\n    set(value) = definedExternally\n    var relatedTarget: EventTarget? /* = null */\n    get() = definedExternally\n    set(value) = definedExternally\n    var region: String? /* = null */\n    get() = definedExternally\n    set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MouseEventInit(screenX: Int? = 0, screenY: Int? = 0, clientX: Int? = 0, clientY: Int? = 0, button: Short? = 0, buttons: Short? = 0, relatedTarget: EventTarget? = null, region: String? = null, ctrlKey: Boolean? = false, shiftKey: Boolean? = false, altKey: Boolean? = false, metaKey: Boolean? = false, modifierAltGraph: Boolean? = false, modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false, modifierFnLock: Boolean? = false, modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false, modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): MouseEventInit {\n    val o = js(\"({})\")\n    o[\"screenX\"] = screenX\n    o[\"screenY\"] = screenY\n    o[\"clientX\"] = clientX\n    o[\"clientY\"] = clientY\n    o[\"button\"] = button\n    o[\"buttons\"] = buttons\n    o[\"relatedTarget\"] = relatedTarget\n    o[\"region\"] = region\n    o[\"ctrlKey\"] = ctrlKey\n    o[\"shiftKey\"] = shiftKey\n    o[\"altKey\"] = altKey\n    o[\"metaKey\"] = metaKey\n    o[\"modifierAltGraph\"] = modifierAltGraph\n    o[\"modifierCapsLock\"] = modifierCapsLock\n    o[\"modifierFn\"] = modifierFn\n    o[\"modifierFnLock\"] = modifierFnLock\n    o[\"modifierHyper\"] = modifierHyper\n    o[\"modifierNumLock\"] = modifierNumLock\n    o[\"modifierScrollLock\"] = modifierScrollLock\n    o[\"modifierSuper\"] = modifierSuper\n    o[\"modifierSymbol\"] = modifierSymbol\n    o[\"modifierSymbolLock\"] = modifierSymbolLock\n    o[\"view\"] = view\n    o[\"detail\"] = detail\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return o\n}\n\npublic external interface EventModifierInit : UIEventInit {\n    var ctrlKey: Boolean? /* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n    var shiftKey: Boolean? /* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n    var altKey: Boolean? /* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n    var metaKey: Boolean? /* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n    var modifierAltGraph: Boolean? /* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n    var modifierCapsLock: Boolean? /* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n    var modifierFn: Boolean? /* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n    var modifierFnLock: Boolean? /* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n    var modifierHyper: Boolean? /* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n    var modifierNumLock: Boolean? /* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n    var modifierScrollLock: Boolean? /* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n    var modifierSuper: Boolean? /* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n    var modifierSymbol: Boolean? /* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n    var modifierSymbolLock: Boolean? /* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun EventModifierInit(ctrlKey: Boolean? = false, shiftKey: Boolean? = false, altKey: Boolean? = false, metaKey: Boolean? = false, modifierAltGraph: Boolean? = false, modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false, modifierFnLock: Boolean? = false, modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false,

```

```

modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view:
Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? =
false): EventModifierInit {\n  val o = js\("\{\}\")\n  o\["ctrlKey"] = ctrlKey\n  o\["shiftKey"] = shiftKey\n
o\["altKey"] = altKey\n  o\["metaKey"] = metaKey\n  o\["modifierAltGraph"] = modifierAltGraph\n
o\["modifierCapsLock"] = modifierCapsLock\n  o\["modifierFn"] = modifierFn\n  o\["modifierFnLock"] =
modifierFnLock\n  o\["modifierHyper"] = modifierHyper\n  o\["modifierNumLock"] = modifierNumLock\n
o\["modifierScrollLock"] = modifierScrollLock\n  o\["modifierSuper"] = modifierSuper\n
o\["modifierSymbol"] = modifierSymbol\n  o\["modifierSymbolLock"] = modifierSymbolLock\n  o\["view"] =
view\n  o\["detail"] = detail\n  o\["bubbles"] = bubbles\n  o\["cancelable"] = cancelable\n  o\["composed"] =
composed\n  return o\n}\n\n/**\n * Exposes the JavaScript

```

```

[WheelEvent](https://developer.mozilla.org/en/docs/Web/API/WheelEvent) to Kotlin\n */\npublic external open
class WheelEvent(type: String, eventInitDict: WheelEventInit = definedExternally) : MouseEvent {\n  open val
deltaX: Double\n  open val deltaY: Double\n  open val deltaZ: Double\n  open val deltaMode: Int\n\n  companion object {\n    val DOM_DELTA_PIXEL: Int\n    val DOM_DELTA_LINE: Int\n    val
DOM_DELTA_PAGE: Int\n    val NONE: Short\n    val CAPTURING_PHASE: Short\n    val
AT_TARGET: Short\n    val BUBBLING_PHASE: Short\n  }\n}\n\npublic external interface WheelEventInit :
MouseEventInit {\n  var deltaX: Double? /* = 0.0 */\n    get() = definedExternally\n    set(value) =
definedExternally\n  var deltaY: Double? /* = 0.0 */\n    get() = definedExternally\n    set(value) =
definedExternally\n  var deltaZ: Double? /* = 0.0 */\n    get() = definedExternally\n    set(value) =
definedExternally\n  var deltaMode: Int? /* = 0 */\n    get() = definedExternally\n    set(value) =
definedExternally\n}\n\n@Suppress\("INVISIBLE_REFERENCE",
\("INVISIBLE_MEMBER")\n)@kotlin.internal.InlineOnly\npublic inline fun WheelEventInit(deltaX: Double? = 0.0,
deltaY: Double? = 0.0, deltaZ: Double? = 0.0, deltaMode: Int? = 0, screenX: Int? = 0, screenY: Int? = 0, clientX:
Int? = 0, clientY: Int? = 0, button: Short? = 0, buttons: Short? = 0, relatedTarget: EventTarget? = null, region:
String? = null, ctrlKey: Boolean? = false, shiftKey: Boolean? = false, altKey: Boolean? = false, metaKey: Boolean?
= false, modifierAltGraph: Boolean? = false, modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false,
modifierFnLock: Boolean? = false, modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false,
modifierScrollLock: Boolean? = false, modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false,
modifierSymbolLock: Boolean? = false, view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false,
cancelable: Boolean? = false, composed: Boolean? = false): WheelEventInit {\n  val o = js\("\{\}\")\n
o\["deltaX"] = deltaX\n  o\["deltaY"] = deltaY\n  o\["deltaZ"] = deltaZ\n  o\["deltaMode"] = deltaMode\n
o\["screenX"] = screenX\n  o\["screenY"] = screenY\n  o\["clientX"] = clientX\n  o\["clientY"] = clientY\n
o\["button"] = button\n  o\["buttons"] = buttons\n  o\["relatedTarget"] = relatedTarget\n  o\["region"] =
region\n  o\["ctrlKey"] = ctrlKey\n  o\["shiftKey"] = shiftKey\n  o\["altKey"] = altKey\n  o\["metaKey"] =
metaKey\n  o\["modifierAltGraph"] = modifierAltGraph\n  o\["modifierCapsLock"] = modifierCapsLock\n
o\["modifierFn"] = modifierFn\n  o\["modifierFnLock"] = modifierFnLock\n  o\["modifierHyper"] =
modifierHyper\n  o\["modifierNumLock"] = modifierNumLock\n  o\["modifierScrollLock"] =
modifierScrollLock\n  o\["modifierSuper"] = modifierSuper\n  o\["modifierSymbol"] = modifierSymbol\n
o\["modifierSymbolLock"] = modifierSymbolLock\n  o\["view"] = view\n  o\["detail"] = detail\n
o\["bubbles"] = bubbles\n  o\["cancelable"] = cancelable\n  o\["composed"] = composed\n  return
o\n}\n\n/**\n * Exposes the JavaScript [InputEvent](https://developer.mozilla.org/en/docs/Web/API/InputEvent) to
Kotlin\n */\npublic external open class InputEvent(type: String, eventInitDict: InputEventInit = definedExternally) :
UIEvent {\n  open val data: String\n  open val isComposing: Boolean\n\n  companion object {\n    val NONE:
Short\n    val CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val BUBBLING_PHASE:
Short\n  }\n}\n\npublic external interface InputEventInit : UIEventInit {\n  var data: String? /* = "" */\n
get() = definedExternally\n  set(value) = definedExternally\n  var isComposing: Boolean? /* = false */\n
get() = definedExternally\n  set(value) = definedExternally\n}\n\n@Suppress\("INVISIBLE_REFERENCE",
\("INVISIBLE_MEMBER")\n)@kotlin.internal.InlineOnly\npublic inline fun InputEventInit(data: String? = "",

```



```

isComposing: Boolean? = false, view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable:
Boolean? = false, composed: Boolean? = false): InputEventInit {\n  val o = js("{}")\n  o["data"] = data\n
o["isComposing"] = isComposing\n  o["view"] = view\n  o["detail"] = detail\n  o["bubbles"] = bubbles\n
o["cancelable"] = cancelable\n  o["composed"] = composed\n  return o\n}\n\n/**\n * Exposes the JavaScript
[KeyboardEvent](https://developer.mozilla.org/en/docs/Web/API/KeyboardEvent) to Kotlin\n *\npublic external
open class KeyboardEvent(type: String, eventInitDict: KeyboardEventInit = definedExternally) : UIEvent {\n
open val key: String\n  open val code: String\n  open val location: Int\n  open val ctrlKey: Boolean\n  open val
shiftKey: Boolean\n  open val altKey: Boolean\n  open val metaKey: Boolean\n  open val repeat: Boolean\n
open val isComposing: Boolean\n  open val charCode: Int\n  open val keyCode: Int\n  open val which: Int\n
fun getModifierState(keyArg: String): Boolean\n\n  companion object {\n    val
DOM_KEY_LOCATION_STANDARD: Int\n    val DOM_KEY_LOCATION_LEFT: Int\n    val
DOM_KEY_LOCATION_RIGHT: Int\n    val DOM_KEY_LOCATION_NUMPAD: Int\n    val NONE:
Short\n    val CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val BUBBLING_PHASE:
Short\n  }\n}\n\npublic external interface KeyboardEventInit : EventModifierInit {\n  var key: String? /* = ""
*/\n  get() = definedExternally\n  set(value) = definedExternally\n  var code: String? /* = "" */\n  get()
= definedExternally\n  set(value) = definedExternally\n  var location: Int? /* = 0 */\n  get() =
definedExternally\n  set(value) = definedExternally\n  var repeat: Boolean? /* = false */\n  get() =
definedExternally\n  set(value) = definedExternally\n  var isComposing: Boolean? /* = false */\n  get() =
definedExternally\n  set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun KeyboardEventInit(key: String? = "",
code: String? = "", location: Int? = 0, repeat: Boolean? = false, isComposing: Boolean? = false, ctrlKey: Boolean?
= false, shiftKey: Boolean? = false, altKey: Boolean? = false, metaKey: Boolean? = false, modifierAltGraph:
Boolean? = false, modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false, modifierFnLock: Boolean? =
false, modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false,
modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view:
Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? =
false): KeyboardEventInit {\n  val o = js("{}")\n  o["key"] = key\n  o["code"] = code\n  o["location"] =
location\n  o["repeat"] = repeat\n  o["isComposing"] = isComposing\n  o["ctrlKey"] = ctrlKey\n
o["shiftKey"] = shiftKey\n  o["altKey"] = altKey\n  o["metaKey"] = metaKey\n  o["modifierAltGraph"] =
modifierAltGraph\n  o["modifierCapsLock"] = modifierCapsLock\n  o["modifierFn"] = modifierFn\n
o["modifierFnLock"] = modifierFnLock\n  o["modifierHyper"] = modifierHyper\n  o["modifierNumLock"] =
modifierNumLock\n  o["modifierScrollLock"] = modifierScrollLock\n  o["modifierSuper"] = modifierSuper\n
o["modifierSymbol"] = modifierSymbol\n  o["modifierSymbolLock"] = modifierSymbolLock\n  o["view"] =
view\n  o["detail"] = detail\n  o["bubbles"] = bubbles\n  o["cancelable"] = cancelable\n  o["composed"] =
composed\n  return o\n}\n\n/**\n * Exposes the JavaScript
[CompositionEvent](https://developer.mozilla.org/en/docs/Web/API/CompositionEvent) to Kotlin\n *\npublic
external open class CompositionEvent(type: String, eventInitDict: CompositionEventInit = definedExternally) :
UIEvent {\n  open val data: String\n\n  companion object {\n    val NONE: Short\n    val
CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val BUBBLING_PHASE: Short\n  }\n}\n\npublic external interface CompositionEventInit : UIEventInit {\n  var data: String? /* = "" */\n  get() =
definedExternally\n  set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun CompositionEventInit(data: String? =
"", view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed:
Boolean? = false): CompositionEventInit {\n  val o = js("{}")\n  o["data"] = data\n  o["view"] = view\n
o["detail"] = detail\n  o["bubbles"] = bubbles\n  o["cancelable"] = cancelable\n  o["composed"] =
composed\n  return o\n}\n\n/**\n * Exposes the JavaScript
[Event](https://developer.mozilla.org/en/docs/Web/API/Event) to Kotlin\n *\npublic external open class
Event(type: String, eventInitDict: EventInit = definedExternally) {\n  open val type: String\n  open val target:

```



```

contextMenu: HTMLMenuElement?
    open var spellcheck: Boolean
    open var innerText: String
    open val offsetParent: Element?
    open val offsetTop: Int
    open val offsetLeft: Int
    open val offsetWidth: Int
    open val offsetHeight: Int
    fun click()
    fun focus()
    fun blur()
    fun forceSpellCheck()
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/** Exposes the JavaScript [HTMLUnknownElement](https://developer.mozilla.org/en/docs/Web/API/HTMLUnknownElement) to Kotlin
 *
 * public external abstract class HTMLUnknownElement : HTMLElement {
 *     companion object {
 *         val ELEMENT_NODE: Short
 *         val ATTRIBUTE_NODE: Short
 *         val TEXT_NODE: Short
 *         val CDATA_SECTION_NODE: Short
 *         val ENTITY_REFERENCE_NODE: Short
 *         val ENTITY_NODE: Short
 *         val PROCESSING_INSTRUCTION_NODE: Short
 *         val COMMENT_NODE: Short
 *         val DOCUMENT_NODE: Short
 *         val DOCUMENT_TYPE_NODE: Short
 *         val DOCUMENT_FRAGMENT_NODE: Short
 *         val NOTATION_NODE: Short
 *         val DOCUMENT_POSITION_DISCONNECTED: Short
 *         val DOCUMENT_POSITION_PRECEDING: Short
 *         val DOCUMENT_POSITION_FOLLOWING: Short
 *         val DOCUMENT_POSITION_CONTAINS: Short
 *         val DOCUMENT_POSITION_CONTAINED_BY: Short
 *         val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
 *     }
 * }

/** Exposes the JavaScript [DOMStringMap](https://developer.mozilla.org/en/docs/Web/API/DOMStringMap) to Kotlin
 *
 * public external abstract class DOMStringMap {
 *     @Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
 *     @kotlin.internal.InlineOnly
 *     public inline operator fun DOMStringMap.get(name: String): String? = asDynamic()[name]
 *     @Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
 *     @kotlin.internal.InlineOnly
 *     public inline operator fun DOMStringMap.set(name: String, value: String) { asDynamic()[name] = value }
 * }

/** Exposes the JavaScript [HTMLHtmlElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHtmlElement) to Kotlin
 *
 * public external abstract class HTMLHtmlElement : HTMLElement {
 *     open var version: String
 *     companion object {
 *         val ELEMENT_NODE: Short
 *         val ATTRIBUTE_NODE: Short
 *         val TEXT_NODE: Short
 *         val CDATA_SECTION_NODE: Short
 *         val ENTITY_REFERENCE_NODE: Short
 *         val ENTITY_NODE: Short
 *         val PROCESSING_INSTRUCTION_NODE: Short
 *         val COMMENT_NODE: Short
 *         val DOCUMENT_NODE: Short
 *         val DOCUMENT_TYPE_NODE: Short
 *         val DOCUMENT_FRAGMENT_NODE: Short
 *         val NOTATION_NODE: Short
 *         val DOCUMENT_POSITION_DISCONNECTED: Short
 *         val DOCUMENT_POSITION_PRECEDING: Short
 *         val DOCUMENT_POSITION_FOLLOWING: Short
 *         val DOCUMENT_POSITION_CONTAINS: Short
 *         val DOCUMENT_POSITION_CONTAINED_BY: Short
 *         val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
 *     }
 * }

/** Exposes the JavaScript [HTMLHeadElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHeadElement) to Kotlin
 *
 * public external abstract class HTMLHeadElement : HTMLElement {
 *     companion object {
 *         val ELEMENT_NODE: Short
 *         val ATTRIBUTE_NODE: Short
 *         val TEXT_NODE: Short
 *         val CDATA_SECTION_NODE: Short
 *         val ENTITY_REFERENCE_NODE: Short
 *         val ENTITY_NODE: Short
 *         val PROCESSING_INSTRUCTION_NODE: Short
 *         val COMMENT_NODE: Short
 *         val DOCUMENT_NODE: Short
 *         val DOCUMENT_TYPE_NODE: Short
 *         val DOCUMENT_FRAGMENT_NODE: Short
 *         val NOTATION_NODE: Short
 *         val DOCUMENT_POSITION_DISCONNECTED: Short
 *         val DOCUMENT_POSITION_PRECEDING: Short
 *         val DOCUMENT_POSITION_FOLLOWING: Short
 *         val DOCUMENT_POSITION_CONTAINS: Short
 *         val DOCUMENT_POSITION_CONTAINED_BY: Short
 *         val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
 *     }
 * }

```

```

    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLTitleElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTitleElement) to Kotlin\n *\npublic
external abstract class HTMLTitleElement : HTMLElement {\n    open var text: String\n\n    companion object {\n
    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val
CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE:
Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLBaseElement](https://developer.mozilla.org/en/docs/Web/API/HTMLBaseElement) to Kotlin\n *\npublic
external abstract class HTMLBaseElement : HTMLElement {\n    open var href: String\n    open var target:
String\n\n    companion object {\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n
    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE:
Short\n    val ENTITY_NODE: Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val
COMMENT_NODE: Short\n    val DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n
    val DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLLinkElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLinkElement) to Kotlin\n *\npublic
external abstract class HTMLLinkElement : HTMLElement, LinkStyle {\n    open var href: String\n    open var
crossOrigin: String?\n    open var rel: String\n    open var `as`: RequestDestination\n    open val relList:
DOMTokenList\n    open var media: String\n    open var nonce: String\n    open var hreflang: String\n    open var
type: String\n    open val sizes: DOMTokenList\n    open varreferrerPolicy: String\n    open var charset: String\n
    open var rev: String\n    open var target: String\n    open var scope: String\n    open var workerType:
WorkerType\n\n    companion object {\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE:
Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val
ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLMetaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMetaElement) to Kotlin\n *\npublic
external abstract class HTMLMetaElement : HTMLElement {\n    open var name: String\n    open var httpEquiv:
String\n    open var content: String\n    open var scheme: String\n\n    companion object {\n    val
ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val
CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE:
Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val

```

```

DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }n}\n\n/**\n * Exposes the JavaScript
[HTMLStyleElement](https://developer.mozilla.org/en/docs/Web/API/HTMLStyleElement) to Kotlin\n *^npublic
external abstract class HTMLStyleElement : HTMLElement, LinkStyle {\n    open var media: String\n    open var
nonce: String\n    open var type: String\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }n}\n\n/**\n * Exposes the JavaScript
[HTMLBodyElement](https://developer.mozilla.org/en/docs/Web/API/HTMLBodyElement) to Kotlin\n *^npublic
external abstract class HTMLBodyElement : HTMLElement, WindowEventHandlers {\n    open var text: String\n
open var link: String\n    open var vLink: String\n    open var aLink: String\n    open var bgColor: String\n
open var background: String\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }n}\n\n/**\n * Exposes the JavaScript
[HTMLHeadingElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHeadingElement) to Kotlin\n
*^npublic external abstract class HTMLHeadingElement : HTMLElement {\n    open var align: String\n\n
companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val
TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE:
Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val
COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n
        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }n}\n\n/**\n * Exposes the JavaScript
[HTMLParagraphElement](https://developer.mozilla.org/en/docs/Web/API/HTMLParagraphElement) to Kotlin\n
*^npublic external abstract class HTMLParagraphElement : HTMLElement {\n    open var align: String\n\n
companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val
TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE:
Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val
COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n
        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val

```

DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript [HTMLHRElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHRElement) to Kotlin \n * \n public external abstract class HTMLHRElement : HTMLElement { \n open var align: String \n open var color: String \n open var noShade: Boolean \n open var size: String \n open var width: String \n \n companion object { \n val ELEMENT_NODE: Short \n val ATTRIBUTE_NODE: Short \n val TEXT_NODE: Short \n val CDATA_SECTION_NODE: Short \n val ENTITY_REFERENCE_NODE: Short \n val ENTITY_NODE: Short \n val PROCESSING_INSTRUCTION_NODE: Short \n val COMMENT_NODE: Short \n val DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE: Short \n val DOCUMENT_FRAGMENT_NODE: Short \n val NOTATION_NODE: Short \n val

DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript [HTMLPreElement](https://developer.mozilla.org/en/docs/Web/API/HTMLPreElement) to Kotlin \n * \n public external abstract class HTMLPreElement : HTMLElement { \n open var width: Int \n \n companion object { \n val ELEMENT_NODE: Short \n val ATTRIBUTE_NODE: Short \n val TEXT_NODE: Short \n val CDATA_SECTION_NODE: Short \n val ENTITY_REFERENCE_NODE: Short \n val ENTITY_NODE: Short \n val PROCESSING_INSTRUCTION_NODE: Short \n val COMMENT_NODE: Short \n val DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE: Short \n val DOCUMENT_FRAGMENT_NODE: Short \n val NOTATION_NODE: Short \n val

DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript [HTMLQuoteElement](https://developer.mozilla.org/en/docs/Web/API/HTMLQuoteElement) to Kotlin \n * \n public external abstract class HTMLQuoteElement : HTMLElement { \n open var cite: String \n \n companion object { \n val ELEMENT_NODE: Short \n val ATTRIBUTE_NODE: Short \n val TEXT_NODE: Short \n val CDATA_SECTION_NODE: Short \n val ENTITY_REFERENCE_NODE: Short \n val ENTITY_NODE: Short \n val PROCESSING_INSTRUCTION_NODE: Short \n val COMMENT_NODE: Short \n val DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE: Short \n val DOCUMENT_FRAGMENT_NODE: Short \n val NOTATION_NODE: Short \n val

DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript [HTMLLOListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLOListElement) to Kotlin \n * \n public external abstract class HTMLLOListElement : HTMLElement { \n open var reversed: Boolean \n open var start: Int \n open var type: String \n open var compact: Boolean \n \n companion object { \n val ELEMENT_NODE: Short \n val ATTRIBUTE_NODE: Short \n val TEXT_NODE: Short \n val CDATA_SECTION_NODE: Short \n val ENTITY_REFERENCE_NODE: Short \n val ENTITY_NODE: Short \n val PROCESSING_INSTRUCTION_NODE: Short \n val COMMENT_NODE: Short \n val DOCUMENT_NODE: Short \n val DOCUMENT_TYPE_NODE: Short \n val DOCUMENT_FRAGMENT_NODE: Short \n val NOTATION_NODE: Short \n val

DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n

```

    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLUListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLUListElement) to Kotlin\n */\npublic
external abstract class HTMLUListElement : HTMLInputElement {\n    open var compact: Boolean\n    open var type:
String\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n
val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE:
Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val
COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n
        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLLIElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLIElement) to Kotlin\n */\npublic
external abstract class HTMLLIElement : HTMLInputElement {\n    open var value: Int\n    open var type: String\n\n
companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val
TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE:
Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val
COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n
        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLDListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDListElement) to Kotlin\n */\npublic
external abstract class HTMLDListElement : HTMLInputElement {\n    open var compact: Boolean\n\n    companion
object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE:
Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val
ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE:
Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLDivElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDivElement) to Kotlin\n */\npublic
external abstract class HTMLDivElement : HTMLInputElement {\n    open var align: String\n\n    companion object {\n
        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLAnchorElement](https://developer.mozilla.org/en/docs/Web/API/HTMLAnchorElement) to Kotlin\n
*/\npublic external abstract class HTMLAnchorElement : HTMLInputElement, HTMLHyperlinkElementUtils {\n    open

```

```

var target: String\n  open var download: String\n  open var ping: String\n  open var rel: String\n  open val
reList: DOMTokenList\n  open var hreflang: String\n  open var type: String\n  open var text: String\n  open
var referrerPolicy: String\n  open var coords: String\n  open var charset: String\n  open var name: String\n
open var rev: String\n  open var shape: String\n\n  companion object {\n    val ELEMENT_NODE: Short\n
val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n
val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n}\n\n/**\n * Exposes the JavaScript
[HTMLDataElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDataElement) to Kotlin\n */\npublic
external abstract class HTMLDataElement : HTMLElement {\n  open var value: String\n\n  companion object {\n
    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val
CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE:
Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n}\n\n/**\n * Exposes the JavaScript
[HTMLTimeElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTimeElement) to Kotlin\n */\npublic
external abstract class HTMLTimeElement : HTMLElement {\n  open var dateTime: String\n\n  companion
object {\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE:
Short\n    val CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val
ENTITY_NODE: Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE:
Short\n    val DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n}\n\n/**\n * Exposes the JavaScript
[HTMLSpanElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSpanElement) to Kotlin\n */\npublic
external abstract class HTMLSpanElement : HTMLElement {\n  companion object {\n    val
ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val
CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE:
Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n}\n\n/**\n * Exposes the JavaScript
[HTMLBRElement](https://developer.mozilla.org/en/docs/Web/API/HTMLBRElement) to Kotlin\n */\npublic
external abstract class HTMLBRElement : HTMLElement {\n  open var clear: String\n\n  companion object {\n

```



```

    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val
CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE:
Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[HTMLHyperlinkElementUtils](https://developer.mozilla.org/en/docs/Web/API/HTMLHyperlinkElementUtils) to
Kotlin \n * \n public external interface HTMLHyperlinkElementUtils { \n    var href: String\n    val origin: String\n
var protocol: String\n    var username: String\n    var password: String\n    var host: String\n    var hostname:
String\n    var port: String\n    var pathname: String\n    var search: String\n    var hash: String \n} \n \n /** \n * Exposes
the JavaScript [HTMLModElement](https://developer.mozilla.org/en/docs/Web/API/HTMLModElement) to
Kotlin \n * \n public external abstract class HTMLModElement : HTMLElement { \n    open var cite: String\n    open
var dateTime: String \n \n    companion object { \n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[HTMLPictureElement](https://developer.mozilla.org/en/docs/Web/API/HTMLPictureElement) to Kotlin \n
* \n public external abstract class HTMLPictureElement : HTMLElement { \n    companion object { \n        val
ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[HTMLSourceElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSourceElement) to Kotlin \n
* \n public external abstract class HTMLSourceElement : HTMLElement { \n    open var src: String\n    open var
type: String\n    open var srcset: String\n    open var sizes: String\n    open var media: String \n \n    companion object
{ \n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n
        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val
ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE:
Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript

```

[HTMLImageElement](https://developer.mozilla.org/en/docs/Web/API/HTMLImageElement) to Kotlin

```

*public external abstract class HTMLImageElement : HTMLElement, HTMLOrSVGImageElement,
    TexImageSource {
    open var alt: String
    open var src: String
    open var srcset: String
    open var sizes: String
    open var crossOrigin: String?
    open var useMap: String
    open var isMap: Boolean
    open var width: Int
    open var height: Int
    open val naturalWidth: Int
    open val naturalHeight: Int
    open val complete: Boolean
    open val currentSrc: String
    open var referrerPolicy: String
    open var name: String
    open var lowsrc: String
    open var align: String
    open var hspace: Int
    open var vspace: Int
    open var longDesc: String
    open var border: String
    open val x: Int
    open val y: Int

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short

        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short

        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

```

* Exposes the JavaScript [HTMLIFrameElement](https://developer.mozilla.org/en/docs/Web/API/HTMLIFrameElement) to Kotlin

```

*public external abstract class HTMLIFrameElement : HTMLElement {
    open var src: String
    open var srcdoc: String
    open var name: String
    open val sandbox: DOMTokenList
    open var allowFullscreen: Boolean
    open var allowUserMedia: Boolean
    open var width: String
    open var height: String
    open var referrerPolicy: String
    open val contentDocument: Document?
    open val contentWindow: Window?
    open var align: String
    open var scrolling: String
    open var frameborder: String
    open var longDesc: String
    open var marginHeight: String
    open var marginWidth: String
    fun getSVGDocument(): Document?

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short

        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short

        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

```

* Exposes the JavaScript [HTMLEmbedElement](https://developer.mozilla.org/en/docs/Web/API/HTMLEmbedElement) to Kotlin

```

*public external abstract class HTMLEmbedElement : HTMLElement {
    open var src: String
    open var type: String
    open var width: String
    open var height: String
    open var align: String
    open var name: String
    fun getSVGDocument(): Document?

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short

        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short

        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

```

* Exposes the JavaScript [HTMLObjectElement](https://developer.mozilla.org/en/docs/Web/API/HTMLObjectElement) to Kotlin

```

*public external abstract class HTMLObjectElement : HTMLElement {
    open var data: String

```

```

type: String\n open var typeMustMatch: Boolean\n open var name: String\n open var useMap: String\n open
val form: HTMLFormElement?\n open var width: String\n open var height: String\n open val
contentDocument: Document?\n open val contentWindow: Window?\n open val willValidate: Boolean\n open
val validity: ValidityState\n open val validationMessage: String\n open var align: String\n open var archive:
String\n open var code: String\n open var declare: Boolean\n open var hspace: Int\n open var standby:
String\n open var vspace: Int\n open var codeBase: String\n open var codeType: String\n open var border:
String\n fun getSVGDocument(): Document?\n fun checkValidity(): Boolean\n fun reportValidity():
Boolean\n fun setCustomValidity(error: String)\n\n companion object {\n val ELEMENT_NODE: Short\n
val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n
val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLParamElement](https://developer.mozilla.org/en/docs/Web/API/HTMLParamElement) to Kotlin\n
*\npublic external abstract class HTMLParamElement : HTMLElement {\n open var name: String\n open var
value: String\n open var type: String\n open var valueType: String\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLVideoElement](https://developer.mozilla.org/en/docs/Web/API/HTMLVideoElement) to Kotlin\n
*\npublic external abstract class HTMLVideoElement : HTMLMediaElement, CanvasImageSource, TexImageSource {\n
open var width: Int\n open var height: Int\n open val videoWidth: Int\n open val videoHeight: Int\n open var
poster: String\n open var playsInline: Boolean\n\n companion object {\n val NETWORK_EMPTY: Short\n
val NETWORK_IDLE: Short\n val NETWORK_LOADING: Short\n val NETWORK_NO_SOURCE:
Short\n val HAVE_NOTHING: Short\n val HAVE_METADATA: Short\n val
HAVE_CURRENT_DATA: Short\n val HAVE_FUTURE_DATA: Short\n val HAVE_ENOUGH_DATA:
Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n
val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLAudioElement](https://developer.mozilla.org/en/docs/Web/API/HTMLAudioElement) to Kotlin\n
*\npublic external abstract class HTMLAudioElement : HTMLMediaElement {\n companion object {\n val
NETWORK_EMPTY: Short\n val NETWORK_IDLE: Short\n val NETWORK_LOADING: Short\n

```

```

val NETWORK_NO_SOURCE: Short\n    val HAVE_NOTHING: Short\n    val HAVE_METADATA:
Short\n    val HAVE_CURRENT_DATA: Short\n    val HAVE_FUTURE_DATA: Short\n    val
HAVE_ENOUGH_DATA: Short\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n
val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE:
Short\n    val ENTITY_NODE: Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val
COMMENT_NODE: Short\n    val DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n
    val DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLTrackElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTrackElement) to Kotlin\n
*/\n\npublic external abstract class HTMLTrackElement : HTMLElement {\n    open var kind: String\n    open var src: String\n
open var srclang: String\n    open var label: String\n    open var default: Boolean\n    open val readyState: Short\n
open val track: TextTrack\n\n    companion object {\n        val NONE: Short\n        val LOADING: Short\n        val
LOADED: Short\n        val ERROR: Short\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE:
Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLMediaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMediaElement) to Kotlin\n
*/\n\npublic external abstract class HTMLMediaElement : HTMLElement {\n    open val error: MediaError?\n    open
var src: String\n    open var srcObject: MediaProvider?\n    open val currentSrc: String\n    open var crossOrigin:
String?\n    open val networkState: Short\n    open var preload: String\n    open val buffered: TimeRanges\n    open
val readyState: Short\n    open val seeking: Boolean\n    open var currentTime: Double\n    open val duration:
Double\n    open val paused: Boolean\n    open var defaultPlaybackRate: Double\n    open var playbackRate:
Double\n    open val played: TimeRanges\n    open val seekable: TimeRanges\n    open val ended: Boolean\n    open
var autoplay: Boolean\n    open var loop: Boolean\n    open var controls: Boolean\n    open var volume: Double\n
open var muted: Boolean\n    open var defaultMuted: Boolean\n    open val audioTracks: AudioTrackList\n    open
val videoTracks: VideoTrackList\n    open val textTracks: TextTrackList\n    open val mediaKeys: MediaKeys?\n
open var onencrypted: ((Event) -> dynamic)?\n    open var onwaitingforkey: ((Event) -> dynamic)?\n    fun load()\n
fun canPlayType(type: String): CanPlayTypeResult\n    fun fastSeek(time: Double)\n    fun getStartDate():
dynamic\n    fun play(): Promise<Unit>\n    fun pause()\n    fun addTextTrack(kind: TextTrackKind, label: String =
definedExternally, language: String = definedExternally): TextTrack\n    fun setMediaKeys(mediaKeys:
MediaKeys?): Promise<Unit>\n\n    companion object {\n        val NETWORK_EMPTY: Short\n        val
NETWORK_IDLE: Short\n        val NETWORK_LOADING: Short\n        val NETWORK_NO_SOURCE: Short\n
        val HAVE_NOTHING: Short\n        val HAVE_METADATA: Short\n        val HAVE_CURRENT_DATA:
Short\n        val HAVE_FUTURE_DATA: Short\n        val HAVE_ENOUGH_DATA: Short\n        val
ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val

```

DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }n}\n\n/**\n * Exposes the JavaScript [MediaError](https://developer.mozilla.org/en/docs/Web/API/MediaError) to Kotlin\n *\npublic external abstract class MediaError {\n open val code: Short\n\n companion object {\n val MEDIA_ERR_ABORTED: Short\n val MEDIA_ERR_NETWORK: Short\n val MEDIA_ERR_DECODE: Short\n val MEDIA_ERR_SRC_NOT_SUPPORTED: Short\n }n}\n\n/**\n * Exposes the JavaScript [AudioTrackList](https://developer.mozilla.org/en/docs/Web/API/AudioTrackList) to Kotlin\n *\npublic external abstract class AudioTrackList : EventTarget {\n open val length: Int\n open var onchange: ((Event) -> dynamic)?\n open var onaddtrack: ((TrackEvent) -> dynamic)?\n open var onremovetrack: ((TrackEvent) -> dynamic)?\n fun getTrackById(id: String): AudioTrack?\n}\n\n@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun AudioTrackList.get(index: Int): AudioTrack? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript [AudioTrack](https://developer.mozilla.org/en/docs/Web/API/AudioTrack) to Kotlin\n *\npublic external abstract class AudioTrack : UnionAudioTrackOrTextTrackOrVideoTrack {\n open val id: String\n open val kind: String\n open val label: String\n open val language: String\n open var enabled: Boolean\n open val sourceBuffer: SourceBuffer?\n}\n\n/**\n * Exposes the JavaScript [VideoTrackList](https://developer.mozilla.org/en/docs/Web/API/VideoTrackList) to Kotlin\n *\npublic external abstract class VideoTrackList : EventTarget {\n open val length: Int\n open val selectedIndex: Int\n open var onchange: ((Event) -> dynamic)?\n open var onaddtrack: ((TrackEvent) -> dynamic)?\n open var onremovetrack: ((TrackEvent) -> dynamic)?\n fun getTrackById(id: String): VideoTrack?\n}\n\n@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun VideoTrackList.get(index: Int): VideoTrack? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript [VideoTrack](https://developer.mozilla.org/en/docs/Web/API/VideoTrack) to Kotlin\n *\npublic external abstract class VideoTrack : UnionAudioTrackOrTextTrackOrVideoTrack {\n open val id: String\n open val kind: String\n open val label: String\n open val language: String\n open var selected: Boolean\n open val sourceBuffer: SourceBuffer?\n}\n\npublic external abstract class TextTrackList : EventTarget {\n open val length: Int\n open var onchange: ((Event) -> dynamic)?\n open var onaddtrack: ((TrackEvent) -> dynamic)?\n open var onremovetrack: ((TrackEvent) -> dynamic)?\n fun getTrackById(id: String): TextTrack?\n}\n\n@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun TextTrackList.get(index: Int): TextTrack? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript [TextTrack](https://developer.mozilla.org/en/docs/Web/API/TextTrack) to Kotlin\n *\npublic external abstract class TextTrack : EventTarget, UnionAudioTrackOrTextTrackOrVideoTrack {\n open val kind: TextTrackKind\n open val label: String\n open val language: String\n open val id: String\n open val inBandMetadataTrackDispatchType: String\n open var mode: TextTrackMode\n open val cues: TextTrackCueList?\n open val activeCues: TextTrackCueList?\n open var oncuechange: ((Event) -> dynamic)?\n open val sourceBuffer: SourceBuffer?\n fun addCue(cue: TextTrackCue)\n fun removeCue(cue: TextTrackCue)\n}\n\npublic external abstract class TextTrackCueList {\n open val length: Int\n fun getCueById(id: String): TextTrackCue?\n}\n\n@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun TextTrackCueList.get(index: Int): TextTrackCue? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript [TextTrackCue](https://developer.mozilla.org/en/docs/Web/API/TextTrackCue) to Kotlin\n *\npublic external abstract class TextTrackCue : EventTarget {\n open val track: TextTrack?\n open var id: String\n open var startTime: Double\n open var endTime: Double\n open var pauseOnExit: Boolean\n open var onenter: ((Event) -> dynamic)?\n open var onexit: ((Event) -> dynamic)?\n}\n\n/**\n * Exposes the JavaScript

[TimeRanges](https://developer.mozilla.org/en/docs/Web/API/TimeRanges) to Kotlin\n *^/npublic external abstract class TimeRanges {\n open val length: Int\n fun start(index: Int): Double\n fun end(index: Int): Double\n}\n/n/**\n * Exposes the JavaScript

[TrackEvent](https://developer.mozilla.org/en/docs/Web/API/TrackEvent) to Kotlin\n *^/npublic external open class TrackEvent(type: String, eventInitDict: TrackEventInit = definedExternally) : Event {\n open val track: UnionAudioTrackOrTextTrackOrVideoTrack?\n companion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n}\n}\n/npublic external interface TrackEventInit : EventInit {\n var track: UnionAudioTrackOrTextTrackOrVideoTrack? /* = null */\n get() = definedExternally\n set(value) = definedExternally\n}\n/n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun TrackEventInit(track: UnionAudioTrackOrTextTrackOrVideoTrack? = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): TrackEventInit {\n val o = js(\"({})\")\n o[\"track\"] = track\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n/n/**\n * Exposes the JavaScript

[HTMLMapElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMapElement) to Kotlin\n *^/npublic external abstract class HTMLMapElement : HTMLElement {\n open var name: String\n open val areas: HTMLCollection\n/n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n/n/**\n * Exposes the JavaScript

[HTMLAreaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLAreaElement) to Kotlin\n *^/npublic external abstract class HTMLAreaElement : HTMLElement, HTMLHyperlinkElementUtils {\n open var alt: String\n open var coords: String\n open var shape: String\n open var target: String\n open var download: String\n open var ping: String\n open var rel: String\n open val relList: DOMTokenList\n open var referrerPolicy: String\n open var noHref: Boolean\n/n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n/n/**\n * Exposes the JavaScript

[HTMLTableElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableElement) to Kotlin\n *^/npublic external abstract class HTMLTableElement : HTMLElement {\n open var caption: HTMLTableCaptionElement?\n open var tHead: HTMLTableSectionElement?\n open var tFoot: HTMLTableSectionElement?\n open val tBodies: HTMLCollection\n open val rows: HTMLCollection\n open var align: String\n open var border: String\n open var frame: String\n open var rules: String\n open var summary: String\n open var width: String\n open var bgColor: String\n open var cellPadding: String\n open var cellSpacing: String\n fun createCaption(): HTMLTableCaptionElement\n fun deleteCaption()\n fun createTHead(): HTMLTableSectionElement\n fun deleteTHead()\n fun createTFoot():


```

*\npublic external abstract class HTMLTableRowElement : HTMLInputElement {\n  open val rowIndex: Int\n  open val sectionRowIndex: Int\n  open val cells: HTMLCollection\n  open var align: String\n  open var ch: String\n  open var chOff: String\n  open var vAlign: String\n  open var bgColor: String\n  fun insertCell(index: Int = definedExternally): HTMLInputElement\n  fun deleteCell(index: Int)\n\n  companion object {\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n}\n\n/**\n * Exposes the JavaScript [HTMLTableCellElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableCellElement) to Kotlin\n */\n\n*\npublic external abstract class HTMLTableCellElement : HTMLInputElement {\n  open var colSpan: Int\n  open var rowspan: Int\n  open var headers: String\n  open val cellIndex: Int\n  open var scope: String\n  open var abbr: String\n  open var align: String\n  open var axis: String\n  open var height: String\n  open var width: String\n  open var ch: String\n  open var chOff: String\n  open var noWrap: Boolean\n  open var vAlign: String\n  open var bgColor: String\n\n  companion object {\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n}\n\n/**\n * Exposes the JavaScript [HTMLFormElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFormElement) to Kotlin\n */\n\n*\npublic external abstract class HTMLFormElement : HTMLInputElement {\n  open var acceptCharset: String\n  open var action: String\n  open var autocomplete: String\n  open var enctype: String\n  open var encoding: String\n  open var method: String\n  open var name: String\n  open var noValidate: Boolean\n  open var target: String\n  open val elements: HTMLFormControlsCollection\n  open val length: Int\n  fun submit()\n  fun reset()\n  fun checkValidity(): Boolean\n  fun reportValidity(): Boolean\n\n  companion object {\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n}\n\n\n@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun HTMLFormElement.get(index: Int): Element? = asDynamic()[index]\n\n\n@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun

```


HTMLFormElement.get(name: String): UnionElementOrRadioNodeList? = asDynamic()[name]\n\n/**\n * Exposes the JavaScript [HTMLLabelElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLabelElement) to Kotlin\n\n */\npublic external abstract class HTMLLabelElement : HTMLInputElement {\n open val form: HTMLFormElement?\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLInputElement](https://developer.mozilla.org/en/docs/Web/API/HTMLInputElement) to Kotlin\n\n */\npublic external abstract class HTMLInputElement : HTMLFormElement {\n open var accept: String\n open var alt: String\n open var autocomplete: String\n open var autofocus: Boolean\n open var defaultChecked: Boolean\n open var checked: Boolean\n open var dirName: String\n open var disabled: Boolean\n open val form: HTMLFormElement?\n open val files: FileList?\n open var formAction: String\n open var formEnctype: String\n open var formMethod: String\n open var formNoValidate: Boolean\n open var formTarget: String\n open var height: Int\n open var indeterminate: Boolean\n open var inputMode: String\n open val list: HTMLListElement?\n open var max: String\n open var maxLength: Int\n open var min: String\n open var minLength: Int\n open var multiple: Boolean\n open var name: String\n open var pattern: String\n open var placeholder: String\n open var readOnly: Boolean\n open var required: Boolean\n open var size: Int\n open var src: String\n open var step: String\n open var type: String\n open var defaultValue: String\n open var value: String\n open var valueAsDate: dynamic\n open var valueAsNumber: Double\n open var width: Int\n open val willValidate: Boolean\n open val validity: ValidityState\n open val validationMessage: String\n open val labels: NodeList\n open var selectionStart: Int?\n open var selectionEnd: Int?\n open var selectionDirection: String?\n open var align: String\n open var useMap: String\n fun stepUp(n: Int = definedExternally)\n fun stepDown(n: Int = definedExternally)\n fun checkValidity(): Boolean\n fun reportValidity(): Boolean\n fun setCustomValidity(error: String)\n fun select()\n fun setRangeText(replacement: String)\n fun setRangeText(replacement: String, start: Int, end: Int, selectionMode: SelectionMode = definedExternally)\n fun setSelectionRange(start: Int, end: Int, direction: String = definedExternally)\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLButtonElement](https://developer.mozilla.org/en/docs/Web/API/HTMLButtonElement) to Kotlin\n\n */\npublic external abstract class HTMLButtonElement : HTMLFormElement {\n open var autofocus: Boolean\n open var disabled: Boolean\n open val form: HTMLFormElement?\n open var formAction: String\n open var formEnctype: String\n open var formMethod: String\n open var formNoValidate: Boolean\n open var formTarget: String\n open var name: String\n open var type: String\n open var value: String\n open var menu: HTMLMenuElement?\n open val willValidate: Boolean\n open val validity: ValidityState\n open val

```

validationMessage: String\n  open val labels: NodeList\n  fun checkValidity(): Boolean\n  fun reportValidity():
Boolean\n  fun setCustomValidity(error: String)\n\n  companion object {\n    val ELEMENT_NODE: Short\n
    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n
    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n}\n\n/**\n * Exposes the JavaScript
[HTMLSelectElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSelectElement) to Kotlin\n
*\n\npublic external abstract class HTMLSelectElement : HTMLElement, ItemArrayLike<Element> {\n  open var
autocomplete: String\n  open var autofocus: Boolean\n  open var disabled: Boolean\n  open val form:
HTMLFormElement?\n  open var multiple: Boolean\n  open var name: String\n  open var required: Boolean\n
open var size: Int\n  open val type: String\n  open val options: HTMLOptionsCollection\n  override var length:
Int\n  open val selectedOptions: HTMLCollection\n  open var selectedIndex: Int\n  open var value: String\n
open val willValidate: Boolean\n  open val validity: ValidityState\n  open val validationMessage: String\n
open val labels: NodeList\n  fun namedItem(name: String): HTMLOptionElement?\n  fun add(element:
UnionHTMLOptGroupElementOrHTMLOptionElement, before: dynamic = definedExternally)\n  fun
remove(index: Int)\n  fun checkValidity(): Boolean\n  fun reportValidity(): Boolean\n  fun
setCustomValidity(error: String)\n  override fun item(index: Int): Element?\n\n  companion object {\n    val
ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val
CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE:
Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun
HTMLSelectElement.get(index: Int): Element? =
asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun
HTMLSelectElement.set(index: Int, option: HTMLOptionElement?) { asDynamic()[index] = option }\n\n/**\n *
Exposes the JavaScript
[HTMLDataListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDataListElement) to Kotlin\n
*\n\npublic external abstract class HTMLDataListElement : HTMLElement {\n  open val options:
HTMLCollection\n\n  companion object {\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE:
Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val
ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n

```

```

    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLOptGroupElement](https://developer.mozilla.org/en/docs/Web/API/HTMLOptGroupElement) to Kotlin\n
*\n\npublic external abstract class HTMLOptGroupElement : HTMLInputElement,
UnionHTMLOptGroupElementOrHTMLOptionElement {\n    open var disabled: Boolean\n    open var label:
String\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n
val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE:
Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val
COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n
val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLOptionElement](https://developer.mozilla.org/en/docs/Web/API/HTMLOptionElement) to Kotlin\n
*\n\npublic external abstract class HTMLOptionElement : HTMLInputElement,
UnionHTMLOptGroupElementOrHTMLOptionElement {\n    open var disabled: Boolean\n    open var form:
HTMLFormElement?\n    open var label: String\n    open var defaultSelected: Boolean\n    open var selected:
Boolean\n    open var value: String\n    open var text: String\n    open val index: Int\n\n    companion object {\n
val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLTextAreaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTextAreaElement) to Kotlin\n
*\n\npublic external abstract class HTMLTextAreaElement : HTMLInputElement {\n    open var autocomplete: String\n
open var autofocus: Boolean\n    open var cols: Int\n    open var dirName: String\n    open var disabled: Boolean\n
open val form: HTMLFormElement?\n    open var inputMode: String\n    open var maxLength: Int\n    open var
minLength: Int\n    open var name: String\n    open var placeholder: String\n    open var readOnly: Boolean\n    open
var required: Boolean\n    open var rows: Int\n    open var wrap: String\n    open val type: String\n    open var
defaultValue: String\n    open var value: String\n    open val textLength: Int\n    open val willValidate: Boolean\n
open val validity: ValidityState\n    open val validationMessage: String\n    open val labels: NodeList\n    open var
selectionStart: Int?\n    open var selectionEnd: Int?\n    open var selectionDirection: String?\n    fun checkValidity():
Boolean\n    fun reportValidity(): Boolean\n    fun setCustomValidity(error: String)\n    fun select()\n    fun
setRangeText(replacement: String)\n    fun setRangeText(replacement: String, start: Int, end: Int, selectionMode:
SelectionMode = definedExternally)\n    fun setSelectionRange(start: Int, end: Int, direction: String =
definedExternally)\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE:
Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n

```

```

val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLKeygenElement](https://developer.mozilla.org/en/docs/Web/API/HTMLKeygenElement) to Kotlin\n
*\n\npublic external abstract class HTMLKeygenElement : HTMLInputElement {\n    open var autofocus: Boolean\n
open var challenge: String\n    open var disabled: Boolean\n    open val form: HTMLFormElement?\n    open var
keytype: String\n    open var name: String\n    open val type: String\n    open val willValidate: Boolean\n    open val
validity: ValidityState\n    open val validationMessage: String\n    open val labels: NodeList\n    fun checkValidity():
Boolean\n    fun reportValidity(): Boolean\n    fun setCustomValidity(error: String)\n\n    companion object {\n
val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLOutputElement](https://developer.mozilla.org/en/docs/Web/API/HTMLOutputElement) to Kotlin\n
*\n\npublic external abstract class HTMLOutputElement : HTMLInputElement {\n    open val htmlFor: DOMTokenList\n
open val form: HTMLFormElement?\n    open var name: String\n    open val type: String\n    open var
defaultValue: String\n    open var value: String\n    open val willValidate: Boolean\n    open val validity:
ValidityState\n    open val validationMessage: String\n    open val labels: NodeList\n    fun checkValidity():
Boolean\n    fun reportValidity(): Boolean\n    fun setCustomValidity(error: String)\n\n    companion object {\n
val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLProgressElement](https://developer.mozilla.org/en/docs/Web/API/HTMLProgressElement) to Kotlin\n
*\n\npublic external abstract class HTMLProgressElement : HTMLInputElement {\n    open var value: Double\n    open
var max: Double\n    open val position: Double\n    open val labels: NodeList\n\n    companion object {\n
val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLMeterElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMeterElement) to Kotlin\n
*\n\npublic external abstract class HTMLMeterElement : HTMLInputElement {\n    open var value: Double\n    open var min:
Double\n    open var max: Double\n    open var low: Double\n    open var high: Double\n    open var optimum:
Double\n    open val labels: NodeList\n\n    companion object {\n
val ELEMENT_NODE: Short\n        val

```

```

ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val
ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[HTMLFieldSetElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFieldSetElement) to Kotlin \n
* \n public external abstract class HTMLFieldSetElement : HTMLElement { \n    open var disabled: Boolean \n
open val form: HTMLFormElement? \n    open var name: String \n    open val type: String \n    open val elements:
HTMLCollection \n    open val willValidate: Boolean \n    open val validity: ValidityState \n    open val
validationMessage: String \n    fun checkValidity(): Boolean \n    fun reportValidity(): Boolean \n    fun
setCustomValidity(error: String) \n    companion object { \n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[HTMLLegendElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLegendElement) to Kotlin \n
* \n public external abstract class HTMLLegendElement : HTMLElement { \n    open val form:
HTMLFormElement? \n    open var align: String \n    companion object { \n        val ELEMENT_NODE: Short\n
        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n
        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[ValidityState](https://developer.mozilla.org/en/docs/Web/API/ValidityState) to Kotlin \n
* \n public external
abstract class ValidityState { \n    open val valueMissing: Boolean \n    open val typeMismatch: Boolean \n    open val
patternMismatch: Boolean \n    open val tooLong: Boolean \n    open val tooShort: Boolean \n    open val
rangeUnderflow: Boolean \n    open val rangeOverflow: Boolean \n    open val stepMismatch: Boolean \n    open val
badInput: Boolean \n    open val customError: Boolean \n    open val valid: Boolean \n} \n \n /** \n * Exposes the
JavaScript [HTMLDetailsElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDetailsElement) to
Kotlin \n
* \n public external abstract class HTMLDetailsElement : HTMLElement { \n    open var open: Boolean \n
companion object { \n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val
TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE:
Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val
COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n
        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val

```

```

DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external abstract class
HTMLMenuElement : HTMLElement {\n    open var type: String\n    open var label: String\n    open var compact:
Boolean\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n
val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE:
Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val
COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n
val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external abstract class
HTMLMenuItemElement : HTMLElement {\n    open var type: String\n    open var label: String\n    open var icon:
String\n    open var disabled: Boolean\n    open var checked: Boolean\n    open var radiogroup: String\n    open var
default: Boolean\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE:
Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external open class
RelatedEvent(type: String, eventInitDict: RelatedEventInit = definedExternally) : Event {\n    open val
relatedTarget: EventTarget?\n\n    companion object {\n        val NONE: Short\n        val CAPTURING_PHASE:
Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }\n}\n\npublic external interface
RelatedEventInit : EventInit {\n    var relatedTarget: EventTarget? /* = null */\n        get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\n\npublic inline fun RelatedEventInit(relatedTarget:
EventTarget? = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false):
RelatedEventInit {\n    val o = js("{}")\n    o["relatedTarget"] = relatedTarget\n    o["bubbles"] = bubbles\n
o["cancelable"] = cancelable\n    o["composed"] = composed\n    return o\n}\n\n/**\n * Exposes the JavaScript
[HTMLDialogElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDialogElement) to Kotlin\n
*/\n\npublic external abstract class HTMLDialogElement : HTMLElement {\n    open var open: Boolean\n    open var
returnValue: String\n    fun show(anchor: UnionElementOrMouseEvent = definedExternally)\n    fun
showModal(anchor: UnionElementOrMouseEvent = definedExternally)\n    fun close(returnValue: String =
definedExternally)\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE:
Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val

```

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLScriptElement](https://developer.mozilla.org/en/docs/Web/API/HTMLScriptElement) to Kotlin\n */\n\npublic external abstract class HTMLScriptElement : HTMLMLElement, HTMLOrSVGScriptElement {\n open var src: String\n open var type: String\n open var charset: String\n open var async: Boolean\n open var defer: Boolean\n open var crossOrigin: String?\n open var text: String\n open var nonce: String\n open var event: String\n open var htmlFor: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLTemplateElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTemplateElement) to Kotlin\n */\n\npublic external abstract class HTMLTemplateElement : HTMLMLElement {\n open val content: DocumentFragment\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLSlotElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSlotElement) to Kotlin\n */\n\npublic external abstract class HTMLSlotElement : HTMLMLElement {\n open var name: String\n fun assignedNodes(options: AssignedNodesOptions = definedExternally): Array<Node>\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external interface AssignedNodesOptions {\n var flatten: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun AssignedNodesOptions(flatten: Boolean? = false): AssignedNodesOptions {\n val o = js(\"({})\")\n o[\"flatten\"] = flatten\n return o\n}\n\n/**\n * Exposes the JavaScript [HTMLCanvasElement](https://developer.mozilla.org/en/docs/Web/API/HTMLCanvasElement) to Kotlin\n */\n\npublic external abstract class HTMLCanvasElement : HTMLMLElement, CanvasImageSource, TexImageSource {\n open var width: Int\n open var height: Int\n fun getContext(contextId: String, vararg arguments: Any?): RenderingContext?\n fun toDataURL(type: String = definedExternally, quality: Any? = definedExternally):

```

String\n fun toBlob(_callback: (Blob?) -> Unit, type: String = definedExternally, quality: Any? =
definedExternally)\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE:
Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external interface
CanvasRenderingContext2DSettings {\n var alpha: Boolean? /* = true */\n get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun
CanvasRenderingContext2DSettings(alpha: Boolean? = true): CanvasRenderingContext2DSettings {\n val o =
js(\"({})\")\n o[\"alpha\"] = alpha\n return o\n}\n\n/**\n * Exposes the JavaScript
[CanvasRenderingContext2D](https://developer.mozilla.org/en/docs/Web/API/CanvasRenderingContext2D) to
Kotlin\n */\n\npublic external abstract class CanvasRenderingContext2D : CanvasState, CanvasTransform,
CanvasCompositing, CanvasImageSmoothing, CanvasFillStrokeStyles, CanvasShadowStyles, CanvasFilters,
CanvasRect, CanvasDrawPath, CanvasUserInterface, CanvasText, CanvasDrawImage, CanvasHitRegion,
CanvasImageData, CanvasPathDrawingStyles, CanvasTextDrawingStyles, CanvasPath, RenderingContext {\n
open val canvas: HTMLCanvasElement\n}\n\npublic external interface CanvasState {\n fun save()\n fun
restore()\n}\n\npublic external interface CanvasTransform {\n fun scale(x: Double, y: Double)\n fun
rotate(angle: Double)\n fun translate(x: Double, y: Double)\n fun transform(a: Double, b: Double, c: Double, d:
Double, e: Double, f: Double)\n fun getTransform(): DOMMatrix\n fun setTransform(a: Double, b: Double, c:
Double, d: Double, e: Double, f: Double)\n fun setTransform(transform: dynamic = definedExternally)\n fun
resetTransform()\n}\n\npublic external interface CanvasCompositing {\n var globalAlpha: Double\n var
globalCompositeOperation: String\n}\n\npublic external interface CanvasImageSmoothing {\n var
imageSmoothingEnabled: Boolean\n var imageSmoothingQuality: ImageSmoothingQuality\n}\n\npublic external
interface CanvasFillStrokeStyles {\n var strokeStyle: dynamic\n get() = definedExternally\n set(value) =
definedExternally\n var fillStyle: dynamic\n get() = definedExternally\n set(value) = definedExternally\n
fun createLinearGradient(x0: Double, y0: Double, x1: Double, y1: Double): CanvasGradient\n fun
createRadialGradient(x0: Double, y0: Double, r0: Double, x1: Double, y1: Double, r1: Double): CanvasGradient\n
fun createPattern(image: CanvasImageSource, repetition: String): CanvasPattern?\n}\n\npublic external interface
CanvasShadowStyles {\n var shadowOffsetX: Double\n var shadowOffsetY: Double\n var shadowBlur:
Double\n var shadowColor: String\n}\n\npublic external interface CanvasFilters {\n var filter:
String\n}\n\npublic external interface CanvasRect {\n fun clearRect(x: Double, y: Double, w: Double, h:
Double)\n fun fillRect(x: Double, y: Double, w: Double, h: Double)\n fun strokeRect(x: Double, y: Double, w:
Double, h: Double)\n}\n\npublic external interface CanvasDrawPath {\n fun beginPath()\n fun fill(fillRule:
CanvasFillRule = definedExternally)\n fun fill(path: Path2D, fillRule: CanvasFillRule = definedExternally)\n
fun stroke()\n fun stroke(path: Path2D)\n fun clip(fillRule: CanvasFillRule = definedExternally)\n fun
clip(path: Path2D, fillRule: CanvasFillRule = definedExternally)\n fun resetClip()\n fun isPointInPath(x:
Double, y: Double, fillRule: CanvasFillRule = definedExternally): Boolean\n fun isPointInPath(path: Path2D, x:
Double, y: Double, fillRule: CanvasFillRule = definedExternally): Boolean\n fun isPointInStroke(x: Double, y:
Double): Boolean\n fun isPointInStroke(path: Path2D, x: Double, y: Double): Boolean\n}\n\npublic external
interface CanvasUserInterface {\n fun drawFocusIfNeeded(element: Element)\n fun drawFocusIfNeeded(path:
Path2D, element: Element)\n fun scrollPathIntoView()\n fun scrollPathIntoView(path: Path2D)\n}\n\npublic
external interface CanvasText {\n fun fillText(text: String, x: Double, y: Double, maxWidth: Double =

```



```

definedExternally)\n fun strokeText(text: String, x: Double, y: Double, maxWidth: Double = definedExternally)\n
fun measureText(text: String): TextMetrics\n}\n\npublic external interface CanvasDrawImage {\n fun
drawImage(image: CanvasImageSource, dx: Double, dy: Double)\n fun drawImage(image: CanvasImageSource,
dx: Double, dy: Double, dw: Double, dh: Double)\n fun drawImage(image: CanvasImageSource, sx: Double, sy:
Double, sw: Double, sh: Double, dx: Double, dy: Double, dw: Double, dh: Double)\n}\n\npublic external interface
CanvasHitRegion {\n fun addHitRegion(options: HitRegionOptions = definedExternally)\n fun
removeHitRegion(id: String)\n fun clearHitRegions()\n}\n\npublic external interface CanvasImageData {\n fun
createImageData(sw: Double, sh: Double): ImageData\n fun createImageData(imagedata: ImageData):
ImageData\n fun getImageData(sx: Double, sy: Double, sw: Double, sh: Double): ImageData\n fun
putImageData(imagedata: ImageData, dx: Double, dy: Double)\n fun putImageData(imagedata: ImageData, dx:
Double, dy: Double, dirtyX: Double, dirtyY: Double, dirtyWidth: Double, dirtyHeight: Double)\n}\n\npublic
external interface CanvasPathDrawingStyles {\n var lineWidth: Double\n var lineCap: CanvasLineCap\n var
lineJoin: CanvasLineJoin\n var miterLimit: Double\n var lineDashOffset: Double\n fun setLineDash(segments:
Array<Double>)\n fun getLineDash(): Array<Double>\n}\n\npublic external interface CanvasTextDrawingStyles
{\n var font: String\n var textAlign: CanvasTextAlign\n var textBaseline: CanvasTextBaseline\n var
direction: CanvasDirection\n}\n\npublic external interface CanvasPath {\n fun closePath()\n fun moveTo(x:
Double, y: Double)\n fun lineTo(x: Double, y: Double)\n fun quadraticCurveTo(cpx: Double, cpy: Double, x:
Double, y: Double)\n fun bezierCurveTo(cp1x: Double, cp1y: Double, cp2x: Double, cp2y: Double, x: Double, y:
Double)\n fun arcTo(x1: Double, y1: Double, x2: Double, y2: Double, radius: Double)\n fun arcTo(x1: Double,
y1: Double, x2: Double, y2: Double, radiusX: Double, radiusY: Double, rotation: Double)\n fun rect(x: Double, y:
Double, w: Double, h: Double)\n fun arc(x: Double, y: Double, radius: Double, startAngle: Double, endAngle:
Double, anticlockwise: Boolean = definedExternally)\n fun ellipse(x: Double, y: Double, radiusX: Double,
radiusY: Double, rotation: Double, startAngle: Double, endAngle: Double, anticlockwise: Boolean =
definedExternally)\n}\n\n/**\n * Exposes the JavaScript
[CanvasGradient](https://developer.mozilla.org/en/docs/Web/API/CanvasGradient) to Kotlin\n */\n\npublic external
abstract class CanvasGradient {\n fun addColorStop(offset: Double, color: String)\n}\n\n/**\n * Exposes the
JavaScript [CanvasPattern](https://developer.mozilla.org/en/docs/Web/API/CanvasPattern) to Kotlin\n */\n\npublic
external abstract class CanvasPattern {\n fun setTransform(transform: dynamic = definedExternally)\n}\n\n/**\n *
Exposes the JavaScript [TextMetrics](https://developer.mozilla.org/en/docs/Web/API/TextMetrics) to Kotlin\n
*/\n\npublic external abstract class TextMetrics {\n open val width: Double\n open val actualBoundingBoxLeft:
Double\n open val actualBoundingBoxRight: Double\n open val fontBoundingBoxAscent: Double\n open val
fontBoundingBoxDescent: Double\n open val actualBoundingBoxAscent: Double\n open val
actualBoundingBoxDescent: Double\n open val emHeightAscent: Double\n open val emHeightDescent:
Double\n open val hangingBaseline: Double\n open val alphabeticBaseline: Double\n open val
ideographicBaseline: Double\n}\n\npublic external interface HitRegionOptions {\n var path: Path2D? /* = null
*/\n fun get() = definedExternally\n fun set(value) = definedExternally\n var fillRule: CanvasFillRule? /* =
CanvasFillRule.NONZERO */\n fun get() = definedExternally\n fun set(value) = definedExternally\n var id:
String? /* = \"\" */\n fun get() = definedExternally\n fun set(value) = definedExternally\n var parentID: String? /*
= null */\n fun get() = definedExternally\n fun set(value) = definedExternally\n var cursor: String? /* = \"inherit\"
*/\n fun get() = definedExternally\n fun set(value) = definedExternally\n var control: Element? /* = null */\n
fun get() = definedExternally\n fun set(value) = definedExternally\n var label: String? /* = null */\n fun get() =
definedExternally\n fun set(value) = definedExternally\n var role: String? /* = null */\n fun get() =
definedExternally\n fun set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun HitRegionOptions(path: Path2D? =
null, fillRule: CanvasFillRule? = CanvasFillRule.NONZERO, id: String? = \"\", parentID: String? = null, cursor:
String? = \"inherit\", control: Element? = null, label: String? = null, role: String? = null): HitRegionOptions {\n
o = js(\"{\}\")\n o[\"path\"] = path\n o[\"fillRule\"] = fillRule\n o[\"id\"] = id\n o[\"parentID\"] = parentID\n
o[\"cursor\"] = cursor\n o[\"control\"] = control\n o[\"label\"] = label\n o[\"role\"] = role\n return

```

```

o\n}\n\n/**\n * Exposes the JavaScript [ImageData](https://developer.mozilla.org/en/docs/Web/API/ImageData) to Kotlin\n * \npublic external open class ImageData : ImageBitmapSource, TexImageSource {\n  constructor(sw: Int, sh: Int)\n  constructor(data: Uint8ClampedArray, sw: Int, sh: Int = definedExternally)\n  open val width: Int\n  open val height: Int\n  open val data: Uint8ClampedArray\n}\n\n/**\n * Exposes the JavaScript [Path2D](https://developer.mozilla.org/en/docs/Web/API/Path2D) to Kotlin\n * \npublic external open class Path2D() : CanvasPath {\n  constructor(path: Path2D)\n  constructor(paths: Array<Path2D>, fillRule: CanvasFillRule = definedExternally)\n  constructor(d: String)\n  fun addPath(path: Path2D, transform: dynamic = definedExternally)\n  override fun closePath()\n  override fun moveTo(x: Double, y: Double)\n  override fun.lineTo(x: Double, y: Double)\n  override fun.quadraticCurveTo(cpx: Double, cpy: Double, x: Double, y: Double)\n  override fun.bezierCurveTo(cp1x: Double, cp1y: Double, cp2x: Double, cp2y: Double, x: Double, y: Double)\n  override fun.arcTo(x1: Double, y1: Double, x2: Double, y2: Double, radius: Double)\n  override fun.arcTo(x1: Double, y1: Double, x2: Double, y2: Double, radiusX: Double, radiusY: Double, rotation: Double)\n  override fun.rect(x: Double, y: Double, w: Double, h: Double)\n  override fun.arc(x: Double, y: Double, radius: Double, startAngle: Double, endAngle: Double, anticlockwise: Boolean /* = definedExternally */) \n  override fun.ellipse(x: Double, y: Double, radiusX: Double, radiusY: Double, rotation: Double, startAngle: Double, endAngle: Double, anticlockwise: Boolean /* = definedExternally */) \n}\n\n/**\n * Exposes the JavaScript [ImageBitmapRenderingContext](https://developer.mozilla.org/en/docs/Web/API/ImageBitmapRenderingContext) to Kotlin\n * \npublic external abstract class ImageBitmapRenderingContext {\n  open val canvas: HTMLCanvasElement\n  fun transferFromImageBitmap(bitmap: ImageBitmap?)\n}\n\npublic external interface ImageBitmapRenderingContextSettings {\n  var alpha: Boolean? /* = true */\n  get() = definedExternally\n  set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ImageBitmapRenderingContextSettings(alpha: Boolean? = true): ImageBitmapRenderingContextSettings {\n  val o = js(\"({})\")\n  o[\"alpha\"] = alpha\n  return o\n}\n\n/**\n * Exposes the JavaScript [CustomElementRegistry](https://developer.mozilla.org/en/docs/Web/API/CustomElementRegistry) to Kotlin\n * \npublic external abstract class CustomElementRegistry {\n  fun define(name: String, constructor: () -> dynamic, options: ElementDefinitionOptions = definedExternally)\n  fun get(name: String): Any?\n  fun whenDefined(name: String): Promise<Unit>\n}\n\npublic external interface ElementDefinitionOptions {\n  var extends: String?\n  get() = definedExternally\n  set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ElementDefinitionOptions(extends: String? = undefined): ElementDefinitionOptions {\n  val o = js(\"({})\")\n  o[\"extends\"] = extends\n  return o\n}\n\npublic external interface ElementContentEditable {\n  var contentEditable: String\n  val isContentEditable: Boolean\n}\n\n/**\n * Exposes the JavaScript [DataTransfer](https://developer.mozilla.org/en/docs/Web/API/DataTransfer) to Kotlin\n * \npublic external abstract class DataTransfer {\n  open var dropEffect: String\n  open var effectAllowed: String\n  open val items: DataTransferItemList\n  open val types: Array<out String>\n  open val files: FileList\n  fun setDragImage(image: Element, x: Int, y: Int)\n  fun getData(format: String): String\n  fun setData(format: String, data: String)\n  fun clearData(format: String = definedExternally)\n}\n\n/**\n * Exposes the JavaScript [DataTransferItemList](https://developer.mozilla.org/en/docs/Web/API/DataTransferItemList) to Kotlin\n * \npublic external abstract class DataTransferItemList {\n  open val length: Int\n  fun add(data: String, type: String): DataTransferItem?\n  fun add(data: File): DataTransferItem?\n  fun remove(index: Int)\n  fun clear()\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun DataTransferItemList.get(index: Int): DataTransferItem? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript [DataTransferItem](https://developer.mozilla.org/en/docs/Web/API/DataTransferItem) to Kotlin\n * \npublic external abstract class DataTransferItem {\n  open val kind: String\n  open val type: String\n  fun getAsString(_callback: ((String) -> Unit)?)\n  fun getAsFile(): File?\n}\n\n/**\n * Exposes the JavaScript

```

```

[DragEvent](https://developer.mozilla.org/en/docs/Web/API/DragEvent) to Kotlin\n */\npublic external open class
DragEvent(type: String, eventInitDict: DragEventInit = definedExternally) : MouseEvent {\n open val
dataTransfer: DataTransfer?\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE:
Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface
DragEventInit : MouseEventInit {\n var dataTransfer: DataTransfer? /* = null */\n get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun DragEventInit(dataTransfer:
DataTransfer? = null, screenX: Int? = 0, screenY: Int? = 0, clientX: Int? = 0, clientY: Int? = 0, button: Short? = 0,
buttons: Short? = 0, relatedTarget: EventTarget? = null, region: String? = null, ctrlKey: Boolean? = false, shiftKey:
Boolean? = false, altKey: Boolean? = false, metaKey: Boolean? = false, modifierAltGraph: Boolean? = false,
modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false, modifierFnLock: Boolean? = false,
modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false,
modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view:
Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? =
false): DragEventInit {\n val o = js("{}")\n o["dataTransfer"] = dataTransfer\n o["screenX"] = screenX\n
o["screenY"] = screenY\n o["clientX"] = clientX\n o["clientY"] = clientY\n o["button"] = button\n
o["buttons"] = buttons\n o["relatedTarget"] = relatedTarget\n o["region"] = region\n o["ctrlKey"] =
ctrlKey\n o["shiftKey"] = shiftKey\n o["altKey"] = altKey\n o["metaKey"] = metaKey\n
o["modifierAltGraph"] = modifierAltGraph\n o["modifierCapsLock"] = modifierCapsLock\n
o["modifierFn"] = modifierFn\n o["modifierFnLock"] = modifierFnLock\n o["modifierHyper"] =
modifierHyper\n o["modifierNumLock"] = modifierNumLock\n o["modifierScrollLock"] =
modifierScrollLock\n o["modifierSuper"] = modifierSuper\n o["modifierSymbol"] = modifierSymbol\n
o["modifierSymbolLock"] = modifierSymbolLock\n o["view"] = view\n o["detail"] = detail\n
o["bubbles"] = bubbles\n o["cancelable"] = cancelable\n o["composed"] = composed\n return
o\n}\n\n/**\n * Exposes the JavaScript [Window](https://developer.mozilla.org/en/docs/Web/API/Window) to
Kotlin\n */\n\npublic external abstract class Window : EventTarget, GlobalEventHandlers, WindowEventHandlers,
WindowOrWorkerGlobalScope, WindowSessionStorage, WindowLocalStorage, GlobalPerformance,
UnionMessagePortOrWindowProxy {\n open val window: Window\n open val self: Window\n open val
document: Document\n open var name: String\n open val location: Location\n open val history: History\n
open val customElements: CustomElementRegistry\n open val locationbar: BarProp\n open val menubar:
BarProp\n open val personalbar: BarProp\n open val scrollbars: BarProp\n open val statusbar: BarProp\n
open val toolbar: BarProp\n open var status: String\n open val closed: Boolean\n open val frames: Window\n
open val length: Int\n open val top: Window\n open var opener: Any?\n open val parent: Window\n open val
frameElement: Element?\n open val navigator: Navigator\n open val applicationCache: ApplicationCache\n
open val external: External\n open val screen: Screen\n open val innerWidth: Int\n open val innerHeight: Int\n
open val scrollX: Double\n open val pageXOffset: Double\n open val scrollY: Double\n open val
pageYOffset: Double\n open val screenX: Int\n open val screenY: Int\n open val outerWidth: Int\n open val
outerHeight: Int\n open val devicePixelRatio: Double\n fun close()\n fun stop()\n fun focus()\n fun blur()\n
fun open(url: String = definedExternally, target: String = definedExternally, features: String = definedExternally):
Window?\n fun alert()\n fun alert(message: String)\n fun confirm(message: String = definedExternally):
Boolean\n fun prompt(message: String = definedExternally, default: String = definedExternally): String?\n fun
print()\n fun requestAnimationFrame(callback: (Double) -> Unit): Int\n fun cancelAnimationFrame(handle:
Int)\n fun postMessage(message: Any?, targetOrigin: String, transfer: Array<dynamic> = definedExternally)\n
fun captureEvents()\n fun releaseEvents()\n fun matchMedia(query: String): MediaQueryList\n fun moveTo(x:
Int, y: Int)\n fun moveBy(x: Int, y: Int)\n fun resizeTo(x: Int, y: Int)\n fun resizeBy(x: Int, y: Int)\n fun
scroll(options: ScrollToOptions = definedExternally)\n fun scroll(x: Double, y: Double)\n fun scrollTo(options:
ScrollToOptions = definedExternally)\n fun scrollTo(x: Double, y: Double)\n fun scrollBy(options:
ScrollToOptions = definedExternally)\n fun scrollBy(x: Double, y: Double)\n fun getComputedStyle(elt:

```

```

Element, pseudoElt: String? = definedExternally):
CSSStyleDeclaration\n}\n\n@Suppress(\\"INVISIBLE_REFERENCE\\",
\\"INVISIBLE_MEMBER\\")\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun Window.get(name: String):
dynamic = asDynamic()[name]\n\n\npublic external abstract class BarProp {\n    open val visible: Boolean\n}\n\n\n/**\n * Exposes the JavaScript [History](https://developer.mozilla.org/en/docs/Web/API/History) to Kotlin\n *\n\n\npublic
external abstract class History {\n    open val length: Int\n    open var scrollRestoration: ScrollRestoration\n    open
val state: Any?\n    fun go(delta: Int = definedExternally)\n    fun back()\n    fun forward()\n    fun pushState(data:
Any?, title: String, url: String? = definedExternally)\n    fun replaceState(data: Any?, title: String, url: String? =
definedExternally)\n}\n\n\n/**\n * Exposes the JavaScript
[Location](https://developer.mozilla.org/en/docs/Web/API/Location) to Kotlin\n *\n\n\npublic external abstract class
Location {\n    open var href: String\n    open val origin: String\n    open var protocol: String\n    open var host:
String\n    open var hostname: String\n    open var port: String\n    open var pathname: String\n    open var search:
String\n    open var hash: String\n    open val ancestorOrigins: Array<out String>\n    fun assign(url: String)\n    fun
replace(url: String)\n    fun reload()\n}\n\n\n/**\n * Exposes the JavaScript
[PopStateEvent](https://developer.mozilla.org/en/docs/Web/API/PopStateEvent) to Kotlin\n *\n\n\npublic external
open class PopStateEvent(type: String, eventInitDict: PopStateEventInit = definedExternally) : Event {\n    open val
state: Any?\n\n    companion object {\n        val NONE: Short\n        val CAPTURING_PHASE: Short\n        val
AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }\n}\n\n\npublic external interface PopStateEventInit
: EventInit {\n    var state: Any? /* = null */\n    get() = definedExternally\n    set(value) =
definedExternally\n}\n\n\n@Suppress(\\"INVISIBLE_REFERENCE\\",
\\"INVISIBLE_MEMBER\\")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun PopStateEventInit(state: Any? = null,
bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): PopStateEventInit {\n    val o
= js(\\"({})\\")\n    o[\"state\"] = state\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n
o[\"composed\"] = composed\n    return o\n}\n\n\n/**\n * Exposes the JavaScript
[HashChangeEvent](https://developer.mozilla.org/en/docs/Web/API/HashChangeEvent) to Kotlin\n *\n\n\npublic
external open class HashChangeEvent(type: String, eventInitDict: HashChangeEventInit = definedExternally) :
Event {\n    open val oldURL: String\n    open val newURL: String\n\n    companion object {\n        val NONE:
Short\n        val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE:
Short\n    }\n}\n\n\npublic external interface HashChangeEventInit : EventInit {\n    var oldURL: String? /* = \\\"\" */\n
get() = definedExternally\n    set(value) = definedExternally\n    var newURL: String? /* = \\\"\" */\n    get() =
definedExternally\n    set(value) = definedExternally\n}\n\n\n@Suppress(\\"INVISIBLE_REFERENCE\\",
\\"INVISIBLE_MEMBER\\")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun HashChangeEventInit(oldURL:
String? = \\\"\", newURL: String? = \\\"\", bubbles: Boolean? = false, cancelable: Boolean? = false, composed:
Boolean? = false): HashChangeEventInit {\n    val o = js(\\"({})\\")\n    o[\"oldURL\"] = oldURL\n    o[\"newURL\"]
= newURL\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n
return o\n}\n\n\n/**\n * Exposes the JavaScript
[PageTransitionEvent](https://developer.mozilla.org/en/docs/Web/API/PageTransitionEvent) to Kotlin\n *\n\n\npublic
external open class PageTransitionEvent(type: String, eventInitDict: PageTransitionEventInit = definedExternally) :
Event {\n    open val persisted: Boolean\n\n    companion object {\n        val NONE: Short\n        val
CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }\n}\n\n\npublic external interface PageTransitionEventInit : EventInit {\n    var persisted: Boolean? /* = false */\n
get() = definedExternally\n    set(value) = definedExternally\n}\n\n\n@Suppress(\\"INVISIBLE_REFERENCE\\",
\\"INVISIBLE_MEMBER\\")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun PageTransitionEventInit(persisted:
Boolean? = false, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false):
PageTransitionEventInit {\n    val o = js(\\"({})\\")\n    o[\"persisted\"] = persisted\n    o[\"bubbles\"] = bubbles\n
o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return o\n}\n\n\n/**\n * Exposes the JavaScript
[BeforeUnloadEvent](https://developer.mozilla.org/en/docs/Web/API/BeforeUnloadEvent) to Kotlin\n *\n\n\npublic
external open class BeforeUnloadEvent : Event {\n    var returnValue: String\n\n    companion object {\n        val

```

```

NONE: Short\n    val CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val
BUBBLING_PHASE: Short\n    }\n}\n\npublic external abstract class ApplicationCache : EventTarget {\n    open
val status: Short\n    open var onchecking: ((Event) -> dynamic)?\n    open var onerror: ((Event) -> dynamic)?\n
open var onnoupdate: ((Event) -> dynamic)?\n    open var ondownloading: ((Event) -> dynamic)?\n    open var
onprogress: ((ProgressEvent) -> dynamic)?\n    open var onupdateready: ((Event) -> dynamic)?\n    open var
oncached: ((Event) -> dynamic)?\n    open var onobsolete: ((Event) -> dynamic)?\n    fun update()\n    fun abort()\n
fun swapCache()\n    companion object {\n        val UNCACHED: Short\n        val IDLE: Short\n        val
CHECKING: Short\n        val DOWNLOADING: Short\n        val UPDATEREADY: Short\n        val OBSOLETE:
Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[NavigatorOnLine](https://developer.mozilla.org/en/docs/Web/API/NavigatorOnLine) to Kotlin\n */\n\npublic
external interface NavigatorOnLine {\n    val onLine: Boolean\n}\n\n/**\n * Exposes the JavaScript
[ErrorEvent](https://developer.mozilla.org/en/docs/Web/API/ErrorEvent) to Kotlin\n */\n\npublic external open class
ErrorEvent(type: String, eventInitDict: ErrorEventInit = definedExternally) : Event {\n    open val message: String\n
open val filename: String\n    open val lineno: Int\n    open val colno: Int\n    open val error: Any?\n\n    companion
object {\n        val NONE: Short\n        val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val
BUBBLING_PHASE: Short\n    }\n}\n\npublic external interface ErrorEventInit : EventInit {\n    var message:
String? /* = \"\" */\n        get() = definedExternally\n        set(value) = definedExternally\n    var filename: String? /*
= \"\" */\n        get() = definedExternally\n        set(value) = definedExternally\n    var lineno: Int? /* = 0 */\n
get() = definedExternally\n        set(value) = definedExternally\n    var colno: Int? /* = 0 */\n        get() =
definedExternally\n        set(value) = definedExternally\n    var error: Any? /* = null */\n        get() =
definedExternally\n        set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ErrorEventInit(message: String? = \"\",
filename: String? = \"\", lineno: Int? = 0, colno: Int? = 0, error: Any? = null, bubbles: Boolean? = false, cancelable:
Boolean? = false, composed: Boolean? = false): ErrorEventInit {\n    val o = js(\"({})\")\n    o[\"message\"] =
message\n    o[\"filename\"] = filename\n    o[\"lineno\"] = lineno\n    o[\"colno\"] = colno\n    o[\"error\"] = error\n
o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return
o\n}\n\n/**\n * Exposes the JavaScript
[PromiseRejectionEvent](https://developer.mozilla.org/en/docs/Web/API/PromiseRejectionEvent) to Kotlin\n */\n\npublic
external open class PromiseRejectionEvent(type: String, eventInitDict: PromiseRejectionEventInit) :
Event {\n    open val promise: Promise<Any?>\n    open val reason: Any?\n\n    companion object {\n        val
NONE: Short\n        val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val
BUBBLING_PHASE: Short\n    }\n}\n\npublic external interface PromiseRejectionEventInit : EventInit {\n    var
promise: Promise<Any?>?\n    var reason: Any?\n        get() = definedExternally\n        set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun PromiseRejectionEventInit(promise:
Promise<Any?>?, reason: Any? = undefined, bubbles: Boolean? = false, cancelable: Boolean? = false, composed:
Boolean? = false): PromiseRejectionEventInit {\n    val o = js(\"({})\")\n    o[\"promise\"] = promise\n
o[\"reason\"] = reason\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] =
composed\n    return o\n}\n\n/**\n * Exposes the JavaScript
[GlobalEventHandlers](https://developer.mozilla.org/en/docs/Web/API/GlobalEventHandlers) to Kotlin\n */\n\npublic
external interface GlobalEventHandlers {\n    var onabort: ((Event) -> dynamic)?\n        get() =
definedExternally\n        set(value) = definedExternally\n    var onblur: ((FocusEvent) -> dynamic)?\n        get() =
definedExternally\n        set(value) = definedExternally\n    var oncancel: ((Event) -> dynamic)?\n        get() =
definedExternally\n        set(value) = definedExternally\n    var oncanplay: ((Event) -> dynamic)?\n        get() =
definedExternally\n        set(value) = definedExternally\n    var oncanplaythrough: ((Event) -> dynamic)?\n        get()
= definedExternally\n        set(value) = definedExternally\n    var onchange: ((Event) -> dynamic)?\n        get() =
definedExternally\n        set(value) = definedExternally\n    var onclick: ((MouseEvent) -> dynamic)?\n        get() =
definedExternally\n        set(value) = definedExternally\n    var onclose: ((Event) -> dynamic)?\n        get() =

```

```

definedExternally\n    set(value) = definedExternally\n    var oncontextmenu: ((MouseEvent) -> dynamic)?\n
get() = definedExternally\n    set(value) = definedExternally\n    var oncuechange: ((Event) -> dynamic)?\n
get() = definedExternally\n    set(value) = definedExternally\n    var ondblclick: ((MouseEvent) -> dynamic)?\n
    get() = definedExternally\n    set(value) = definedExternally\n    var ondrag: ((DragEvent) -> dynamic)?\n
get() = definedExternally\n    set(value) = definedExternally\n    var ondragend: ((DragEvent) -> dynamic)?\n
get() = definedExternally\n    set(value) = definedExternally\n    var ondragenter: ((DragEvent) -> dynamic)?\n
    get() = definedExternally\n    set(value) = definedExternally\n    var ondragexit: ((DragEvent) -> dynamic)?\n
get() = definedExternally\n    set(value) = definedExternally\n    var ondragleave: ((DragEvent) -> dynamic)?\n
    get() = definedExternally\n    set(value) = definedExternally\n    var ondragover: ((DragEvent) -> dynamic)?\n
    get() = definedExternally\n    set(value) = definedExternally\n    var ondragstart: ((DragEvent) -> dynamic)?\n
    get() = definedExternally\n    set(value) = definedExternally\n    var ondrop: ((DragEvent) -> dynamic)?\n
get() = definedExternally\n    set(value) = definedExternally\n    var ondurationchange: ((Event) -> dynamic)?\n
    get() = definedExternally\n    set(value) = definedExternally\n    var onemptied: ((Event) -> dynamic)?\n
get() = definedExternally\n    set(value) = definedExternally\n    var onended: ((Event) -> dynamic)?\n    get() =
definedExternally\n    set(value) = definedExternally\n    var onerror: ((dynamic, String, Int, Int, Any?) ->
dynamic)?\n    get() = definedExternally\n    set(value) = definedExternally\n    var onfocus: ((FocusEvent) ->
dynamic)?\n    get() = definedExternally\n    set(value) = definedExternally\n    var oninput: ((InputEvent) ->
dynamic)?\n    get() = definedExternally\n    set(value) = definedExternally\n    var oninvalid: ((Event) ->
dynamic)?\n    get() = definedExternally\n    set(value) = definedExternally\n    var onkeydown:
((KeyboardEvent) -> dynamic)?\n    get() = definedExternally\n    set(value) = definedExternally\n    var
onkeypress: ((KeyboardEvent) -> dynamic)?\n    get() = definedExternally\n    set(value) = definedExternally\n
var onkeyup: ((KeyboardEvent) -> dynamic)?\n    get() = definedExternally\n    set(value) =
definedExternally\n    var onload: ((Event) -> dynamic)?\n    get() = definedExternally\n    set(value) =
definedExternally\n    var onloadeddata: ((Event) -> dynamic)?\n    get() = definedExternally\n    set(value) =
definedExternally\n    var onloadedmetadata: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onloadend: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onloadstart: ((ProgressEvent) -> dynamic)?\n    get() = definedExternally\n
    set(value) = definedExternally\n    var onmousedown: ((MouseEvent) -> dynamic)?\n    get() =
definedExternally\n    set(value) = definedExternally\n    var onmouseenter: ((MouseEvent) -> dynamic)?\n
get() = definedExternally\n    set(value) = definedExternally\n    var onmouseleave: ((MouseEvent) ->
dynamic)?\n    get() = definedExternally\n    set(value) = definedExternally\n    var onmousemove:
((MouseEvent) -> dynamic)?\n    get() = definedExternally\n    set(value) = definedExternally\n    var
onmouseout: ((MouseEvent) -> dynamic)?\n    get() = definedExternally\n    set(value) = definedExternally\n
var onmouseover: ((MouseEvent) -> dynamic)?\n    get() = definedExternally\n    set(value) =
definedExternally\n    var onmouseup: ((MouseEvent) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onwheel: ((WheelEvent) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onpause: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onplay: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onplaying: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onprogress: ((ProgressEvent) -> dynamic)?\n    get() = definedExternally\n
    set(value) = definedExternally\n    var onratechange: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onreset: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onresize: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onscroll: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onseeked: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onseeking: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onselect: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onshow: ((Event) -> dynamic)?\n    get() = definedExternally\n

```

```

set(value) = definedExternally\n    var onstalled: ((Event) -> dynamic)?\n        get() = definedExternally\n
set(value) = definedExternally\n    var onsubmit: ((Event) -> dynamic)?\n        get() = definedExternally\n
set(value) = definedExternally\n    var onsuspend: ((Event) -> dynamic)?\n        get() = definedExternally\n
set(value) = definedExternally\n    var ontimeupdate: ((Event) -> dynamic)?\n        get() = definedExternally\n
set(value) = definedExternally\n    var ontoggle: ((Event) -> dynamic)?\n        get() = definedExternally\n
set(value) = definedExternally\n    var onvolumechange: ((Event) -> dynamic)?\n        get() = definedExternally\n
set(value) = definedExternally\n    var onwaiting: ((Event) -> dynamic)?\n        get() = definedExternally\n
set(value) = definedExternally\n    var ongotpointercapture: ((PointerEvent) -> dynamic)?\n        get() =
definedExternally\n    set(value) = definedExternally\n    var onlostpointercapture: ((PointerEvent) -> dynamic)?\n
    get() = definedExternally\n    set(value) = definedExternally\n    var onpointerdown: ((PointerEvent) ->
dynamic)?\n    get() = definedExternally\n    set(value) = definedExternally\n    var onpointermove:
((PointerEvent) -> dynamic)?\n    get() = definedExternally\n    set(value) = definedExternally\n    var
onpointerup: ((PointerEvent) -> dynamic)?\n    get() = definedExternally\n    set(value) = definedExternally\n
var onpointercancel: ((PointerEvent) -> dynamic)?\n    get() = definedExternally\n    set(value) =
definedExternally\n    var onpointerover: ((PointerEvent) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onpointerout: ((PointerEvent) -> dynamic)?\n    get() = definedExternally\n
    set(value) = definedExternally\n    var onpointerenter: ((PointerEvent) -> dynamic)?\n    get() =
definedExternally\n    set(value) = definedExternally\n    var onpointerleave: ((PointerEvent) -> dynamic)?\n
get() = definedExternally\n    set(value) = definedExternally\n}\n\n**\n * Exposes the JavaScript
[WindowEventHandlers](https://developer.mozilla.org/en/docs/Web/API/WindowEventHandlers) to Kotlin\n
*\npublic external interface WindowEventHandlers {\n    var onafterprint: ((Event) -> dynamic)?\n        get() =
definedExternally\n        set(value) = definedExternally\n    var onbeforeprint: ((Event) -> dynamic)?\n        get() =
definedExternally\n        set(value) = definedExternally\n    var onbeforeunload: ((BeforeUnloadEvent) ->
String)?\n        get() = definedExternally\n        set(value) = definedExternally\n    var onhashchange:
((HashChangeEvent) -> dynamic)?\n        get() = definedExternally\n        set(value) = definedExternally\n    var
onlanguagechange: ((Event) -> dynamic)?\n        get() = definedExternally\n        set(value) = definedExternally\n
var onmessage: ((MessageEvent) -> dynamic)?\n        get() = definedExternally\n        set(value) =
definedExternally\n    var onoffline: ((Event) -> dynamic)?\n        get() = definedExternally\n        set(value) =
definedExternally\n    var ononline: ((Event) -> dynamic)?\n        get() = definedExternally\n        set(value) =
definedExternally\n    var onpagehide: ((PageTransitionEvent) -> dynamic)?\n        get() = definedExternally\n
set(value) = definedExternally\n    var onpageshow: ((PageTransitionEvent) -> dynamic)?\n        get() =
definedExternally\n        set(value) = definedExternally\n    var onpopstate: ((PopStateEvent) -> dynamic)?\n
get() = definedExternally\n        set(value) = definedExternally\n    var onrejectionhandled: ((Event) -> dynamic)?\n
    get() = definedExternally\n        set(value) = definedExternally\n    var onstorage: ((StorageEvent) -> dynamic)?\n
    get() = definedExternally\n        set(value) = definedExternally\n    var onunhandledrejection:
((PromiseRejectionEvent) -> dynamic)?\n        get() = definedExternally\n        set(value) = definedExternally\n
var onunload: ((Event) -> dynamic)?\n        get() = definedExternally\n        set(value) =
definedExternally\n}\n\npublic external interface DocumentAndElementEventHandlers {\n    var oncopy:
((ClipboardEvent) -> dynamic)?\n        get() = definedExternally\n        set(value) = definedExternally\n    var oncut:
((ClipboardEvent) -> dynamic)?\n        get() = definedExternally\n        set(value) = definedExternally\n    var
onpaste: ((ClipboardEvent) -> dynamic)?\n        get() = definedExternally\n        set(value) =
definedExternally\n}\n\n**\n * Exposes the JavaScript
[WindowOrWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/WindowOrWorkerGlobalScope)
to Kotlin\n
*\npublic external interface WindowOrWorkerGlobalScope {\n    val origin: String\n    val caches:
CacheStorage\n    fun btoa(data: String): String\n    fun atob(data: String): String\n    fun setTimeout(handler:
dynamic, timeout: Int = definedExternally, vararg arguments: Any?): Int\n    fun clearTimeout(handle: Int =
definedExternally)\n    fun setInterval(handler: dynamic, timeout: Int = definedExternally, vararg arguments: Any?):
Int\n    fun clearInterval(handle: Int = definedExternally)\n    fun createImageBitmap(image: ImageBitmapSource,

```

```

options: ImageBitmapOptions = definedExternally): Promise<ImageBitmap>\n fun createImageBitmap(image:
ImageBitmapSource, sx: Int, sy: Int, sw: Int, sh: Int, options: ImageBitmapOptions = definedExternally):
Promise<ImageBitmap>\n fun fetch(input: dynamic, init: RequestInit = definedExternally):
Promise<Response>\n}\n\n/**\n * Exposes the JavaScript
[Navigator](https://developer.mozilla.org/en/docs/Web/API/Navigator) to Kotlin\n *\npublic external abstract class
Navigator : NavigatorID, NavigatorLanguage, NavigatorOnLine, NavigatorContentUtils, NavigatorCookies,
NavigatorPlugins, NavigatorConcurrentHardware {\n open val clipboard: Clipboard\n open val mediaDevices:
MediaDevices\n open val maxTouchPoints: Int\n open val serviceWorker: ServiceWorkerContainer\n fun
requestMediaKeySystemAccess(keySystem: String, supportedConfigurations:
Array<MediaKeySystemConfiguration>): Promise<MediaKeySystemAccess>\n fun getUserMedia(constraints:
MediaStreamConstraints, successCallback: (MediaStream) -> Unit, errorCallback: (dynamic) -> Unit)\n fun
vibrate(pattern: dynamic): Boolean\n}\n\n/**\n * Exposes the JavaScript
[NavigatorID](https://developer.mozilla.org/en/docs/Web/API/NavigatorID) to Kotlin\n *\npublic external interface
NavigatorID {\n val appCodeName: String\n val appName: String\n val appVersion: String\n val platform:
String\n val product: String\n val productSub: String\n val userAgent: String\n val vendor: String\n val
vendorSub: String\n val oscpu: String\n fun taintEnabled(): Boolean\n}\n\n/**\n * Exposes the JavaScript
[NavigatorLanguage](https://developer.mozilla.org/en/docs/Web/API/NavigatorLanguage) to Kotlin\n *\npublic
external interface NavigatorLanguage {\n val language: String\n val languages: Array<out String>\n}\n\npublic
external interface NavigatorContentUtils {\n fun registerProtocolHandler(scheme: String, url: String, title:
String)\n fun registerContentHandler(mimeType: String, url: String, title: String)\n fun
isProtocolHandlerRegistered(scheme: String, url: String): String\n fun isContentHandlerRegistered(mimeType:
String, url: String): String\n fun unregisterProtocolHandler(scheme: String, url: String)\n fun
unregisterContentHandler(mimeType: String, url: String)\n}\n\npublic external interface NavigatorCookies {\n val
cookieEnabled: Boolean\n}\n\n/**\n * Exposes the JavaScript
[NavigatorPlugins](https://developer.mozilla.org/en/docs/Web/API/NavigatorPlugins) to Kotlin\n *\npublic
external interface NavigatorPlugins {\n val plugins: PluginArray\n val mimeTypes: MimeTypeArray\n fun
javaEnabled(): Boolean\n}\n\n/**\n * Exposes the JavaScript
[PluginArray](https://developer.mozilla.org/en/docs/Web/API/PluginArray) to Kotlin\n *\npublic external abstract
class PluginArray : ItemArrayLike<Plugin> {\n fun refresh(reload: Boolean = definedExternally)\n override fun
item(index: Int): Plugin?\n fun namedItem(name: String):
Plugin?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun PluginArray.get(index: Int):
Plugin? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun PluginArray.get(name:
String): Plugin? = asDynamic()[name]\n\n/**\n * Exposes the JavaScript
[MimeTypeArray](https://developer.mozilla.org/en/docs/Web/API/MimeTypeArray) to Kotlin\n *\npublic external
abstract class MimeTypeArray : ItemArrayLike<MimeType> {\n override fun item(index: Int): MimeType?\n
fun namedItem(name: String): MimeType?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun MimeTypeArray.get(index:
Int): MimeType? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun MimeTypeArray.get(name:
String): MimeType? = asDynamic()[name]\n\n/**\n * Exposes the JavaScript
[Plugin](https://developer.mozilla.org/en/docs/Web/API/Plugin) to Kotlin\n *\npublic external abstract class Plugin
: ItemArrayLike<MimeType> {\n open val name: String\n open val description: String\n open val filename:
String\n override fun item(index: Int): MimeType?\n fun namedItem(name: String):
MimeType?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Plugin.get(index: Int):
MimeType? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",

```



```

\`INVISIBLE_MEMBER\`)\n@kotlin.internal.InlineOnly\npublic inline operator fun Plugin.get(name: String):
MimeType? = asDynamic()[name]\n\n/**\n * Exposes the JavaScript
[MimeType](https://developer.mozilla.org/en/docs/Web/API/MimeType) to Kotlin\n *\npublic external abstract
class MimeType {\n  open val type: String\n  open val description: String\n  open val suffixes: String\n  open
val enabledPlugin: Plugin\n}\n\n/**\n * Exposes the JavaScript
[ImageBitmap](https://developer.mozilla.org/en/docs/Web/API/ImageBitmap) to Kotlin\n *\npublic external
abstract class ImageBitmap : CanvasImageSource, TexImageSource {\n  open val width: Int\n  open val height:
Int\n  fun close()\n}\n\npublic external interface ImageBitmapOptions {\n  var imageOrientation:
ImageOrientation? /* = ImageOrientation.NONE */\n  get() = definedExternally\n  set(value) =
definedExternally\n  var premultiplyAlpha: PremultiplyAlpha? /* = PremultiplyAlpha.DEFAULT */\n  get() =
definedExternally\n  set(value) = definedExternally\n  var colorSpaceConversion: ColorSpaceConversion? /* =
ColorSpaceConversion.DEFAULT */\n  get() = definedExternally\n  set(value) = definedExternally\n  var
resizeWidth: Int?\n  get() = definedExternally\n  set(value) = definedExternally\n  var resizeHeight: Int?\n
  get() = definedExternally\n  set(value) = definedExternally\n  var resizeQuality: ResizeQuality? /* =
ResizeQuality.LOW */\n  get() = definedExternally\n  set(value) =
definedExternally\n}\n\n@Suppress(\`INVISIBLE_REFERENCE\`,
\`INVISIBLE_MEMBER\`)\n@kotlin.internal.InlineOnly\npublic inline fun
ImageBitmapOptions(imageOrientation: ImageOrientation? = ImageOrientation.NONE, premultiplyAlpha:
PremultiplyAlpha? = PremultiplyAlpha.DEFAULT, colorSpaceConversion: ColorSpaceConversion? =
ColorSpaceConversion.DEFAULT, resizeWidth: Int? = undefined, resizeHeight: Int? = undefined, resizeQuality:
ResizeQuality? = ResizeQuality.LOW): ImageBitmapOptions {\n  val o = js(\`({})\`)\n  o["imageOrientation"]
= imageOrientation\n  o["premultiplyAlpha"] = premultiplyAlpha\n  o["colorSpaceConversion"] =
colorSpaceConversion\n  o["resizeWidth"] = resizeWidth\n  o["resizeHeight"] = resizeHeight\n
o["resizeQuality"] = resizeQuality\n  return o\n}\n\n/**\n * Exposes the JavaScript
[MessageEvent](https://developer.mozilla.org/en/docs/Web/API/MessageEvent) to Kotlin\n *\npublic external open
class MessageEvent(type: String, eventInitDict: MessageEventInit = definedExternally) : Event {\n  open val data:
Any?\n  open val origin: String\n  open val lastEventId: String\n  open val source:
UnionMessagePortOrWindowProxy?\n  open val ports: Array<out MessagePort>\n  fun initMessageEvent(type:
String, bubbles: Boolean, cancelable: Boolean, data: Any?, origin: String, lastEventId: String, source:
UnionMessagePortOrWindowProxy?, ports: Array<MessagePort>)\n\n  companion object {\n    val NONE:
Short\n    val CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val BUBBLING_PHASE:
Short\n  }\n}\n\npublic external interface MessageEventInit : EventInit {\n  var data: Any? /* = null */\n  get()
= definedExternally\n  set(value) = definedExternally\n  var origin: String? /* = \"\" */\n  get() =
definedExternally\n  set(value) = definedExternally\n  var lastEventId: String? /* = \"\" */\n  get() =
definedExternally\n  set(value) = definedExternally\n  var source: UnionMessagePortOrWindowProxy? /* =
null */\n  get() = definedExternally\n  set(value) = definedExternally\n  var ports: Array<MessagePort>? /*
= arrayOf() */\n  get() = definedExternally\n  set(value) =
definedExternally\n}\n\n@Suppress(\`INVISIBLE_REFERENCE\`,
\`INVISIBLE_MEMBER\`)\n@kotlin.internal.InlineOnly\npublic inline fun MessageEventInit(data: Any? = null,
origin: String? = \"\", lastEventId: String? = \"\", source: UnionMessagePortOrWindowProxy? = null, ports:
Array<MessagePort>? = arrayOf(), bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? =
false): MessageEventInit {\n  val o = js(\`({})\`)\n  o["data"] = data\n  o["origin"] = origin\n
o["lastEventId"] = lastEventId\n  o["source"] = source\n  o["ports"] = ports\n  o["bubbles"] = bubbles\n
o["cancelable"] = cancelable\n  o["composed"] = composed\n  return o\n}\n\n/**\n * Exposes the JavaScript
[EventSource](https://developer.mozilla.org/en/docs/Web/API/EventSource) to Kotlin\n *\npublic external open
class EventSource(url: String, eventSourceInitDict: EventSourceInit = definedExternally) : EventTarget {\n  open
val url: String\n  open val withCredentials: Boolean\n  open val readyState: Short\n  var onopen: ((Event) ->
dynamic)?\n  var onmessage: ((MessageEvent) -> dynamic)?\n  var onerror: ((Event) -> dynamic)?\n  fun

```

```

close()\n\n companion object {\n    val CONNECTING: Short\n    val OPEN: Short\n    val CLOSED: Short\n } }\n\npublic external interface EventSourceInit {\n    var withCredentials: Boolean? /* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n }\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun EventSourceInit(withCredentials: Boolean? = false): EventSourceInit {\n    val o = js(\"({})\")\n    o[\"withCredentials\"] = withCredentials\n    return o\n }\n\n/**\n * Exposes the JavaScript [WebSocket](https://developer.mozilla.org/en/docs/Web/API/WebSocket) to Kotlin\n */\npublic external open class WebSocket(url: String, protocols: dynamic = definedExternally) : EventTarget {\n    open val url: String\n    open val readyState: Short\n    open val bufferedAmount: Number\n    var onopen: ((Event) -> dynamic)?\n    var onerror: ((Event) -> dynamic)?\n    var onclose: ((Event) -> dynamic)?\n    open val extensions: String\n    open val protocol: String\n    var onmessage: ((MessageEvent) -> dynamic)?\n    var binaryType: BinaryType\n    fun close(code: Short = definedExternally, reason: String = definedExternally)\n    fun send(data: String)\n    fun send(data: Blob)\n    fun send(data: ArrayBuffer)\n    fun send(data: ArrayBufferView)\n\n companion object {\n    val CONNECTING: Short\n    val OPEN: Short\n    val CLOSING: Short\n    val CLOSED: Short\n } }\n\n/**\n * Exposes the JavaScript [CloseEvent](https://developer.mozilla.org/en/docs/Web/API/CloseEvent) to Kotlin\n */\npublic external open class CloseEvent(type: String, eventInitDict: CloseEventInit = definedExternally) : Event {\n    open val wasClean: Boolean\n    open val code: Short\n    open val reason: String\n\n companion object {\n    val NONE: Short\n    val CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val BUBBLING_PHASE: Short\n } }\n\npublic external interface CloseEventInit : EventInit {\n    var wasClean: Boolean? /* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n    var code: Short? /* = 0 */\n    get() = definedExternally\n    set(value) = definedExternally\n    var reason: String? /* = \"\" */\n    get() = definedExternally\n    set(value) = definedExternally\n }\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun CloseEventInit(wasClean: Boolean? = false, code: Short? = 0, reason: String? = \"\", bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): CloseEventInit {\n    val o = js(\"({})\")\n    o[\"wasClean\"] = wasClean\n    o[\"code\"] = code\n    o[\"reason\"] = reason\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return o\n }\n\n/**\n * Exposes the JavaScript [MessageChannel](https://developer.mozilla.org/en/docs/Web/API/MessageChannel) to Kotlin\n */\npublic external open class MessageChannel {\n    open val port1: MessagePort\n    open val port2: MessagePort\n }\n\n/**\n * Exposes the JavaScript [MessagePort](https://developer.mozilla.org/en/docs/Web/API/MessagePort) to Kotlin\n */\npublic external abstract class MessagePort : EventTarget, UnionMessagePortOrWindowProxy, UnionMessagePortOrServiceWorker, UnionClientOrMessagePortOrServiceWorker {\n    open var onmessage: ((MessageEvent) -> dynamic)?\n    fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)\n    fun start()\n    fun close()\n }\n\n/**\n * Exposes the JavaScript [BroadcastChannel](https://developer.mozilla.org/en/docs/Web/API/BroadcastChannel) to Kotlin\n */\npublic external open class BroadcastChannel(name: String) : EventTarget {\n    open val name: String\n    var onmessage: ((MessageEvent) -> dynamic)?\n    fun postMessage(message: Any?)\n    fun close()\n }\n\n/**\n * Exposes the JavaScript [WorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/WorkerGlobalScope) to Kotlin\n */\npublic external abstract class WorkerGlobalScope : EventTarget, WindowOrWorkerGlobalScope, GlobalPerformance {\n    open val self: WorkerGlobalScope\n    open val location: WorkerLocation\n    open val navigator: WorkerNavigator\n    open var onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)?\n    open var onlanguagechange: ((Event) -> dynamic)?\n    open var onoffline: ((Event) -> dynamic)?\n    open var ononline: ((Event) -> dynamic)?\n    open var onrejectionhandled: ((Event) -> dynamic)?\n    open var onunhandledrejection: ((PromiseRejectionEvent) -> dynamic)?\n    fun importScripts(vararg urls: String)\n }\n\n/**\n * Exposes the JavaScript [DedicatedWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/DedicatedWorkerGlobalScope) to Kotlin\n */\npublic external abstract class DedicatedWorkerGlobalScope : WorkerGlobalScope {\n    open var onmessage: ((MessageEvent) -> dynamic)?\n    fun postMessage(message: Any?, transfer: Array<dynamic> =

```

```

definedExternally)\n fun close()\n}\n\n/**\n * Exposes the JavaScript
[SharedWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/SharedWorkerGlobalScope) to Kotlin\n *\npublic external abstract class SharedWorkerGlobalScope : WorkerGlobalScope {\n open val name:
String\n open val applicationCache: ApplicationCache\n open var onconnect: ((Event) -> dynamic)?\n fun
close()\n}\n\n/**\n * Exposes the JavaScript
[AbstractWorker](https://developer.mozilla.org/en/docs/Web/API/AbstractWorker) to Kotlin\n *\npublic external
interface AbstractWorker {\n var onerror: ((Event) -> dynamic)?\n get() = definedExternally\n set(value)
= definedExternally\n}\n\n/**\n * Exposes the JavaScript
[Worker](https://developer.mozilla.org/en/docs/Web/API/Worker) to Kotlin\n *\npublic external open class
Worker(scriptURL: String, options: WorkerOptions = definedExternally) : EventTarget, AbstractWorker {\n var
onmessage: ((MessageEvent) -> dynamic)?\n override var onerror: ((Event) -> dynamic)?\n fun terminate()\n
fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)\n}\n\npublic external interface
WorkerOptions {\n var type: WorkerType? /* = WorkerType.CLASSIC *\n get() = definedExternally\n
set(value) = definedExternally\n var credentials: RequestCredentials? /* = RequestCredentials.OMIT *\n
get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun WorkerOptions(type: WorkerType? =
WorkerType.CLASSIC, credentials: RequestCredentials? = RequestCredentials.OMIT): WorkerOptions {\n val o
= js(\"({})\")\n o[\"type\"] = type\n o[\"credentials\"] = credentials\n return o\n}\n\n/**\n * Exposes the
JavaScript [SharedWorker](https://developer.mozilla.org/en/docs/Web/API/SharedWorker) to Kotlin\n *\npublic
external open class SharedWorker(scriptURL: String, name: String = definedExternally, options: WorkerOptions =
definedExternally) : EventTarget, AbstractWorker {\n open val port: MessagePort\n override var onerror:
((Event) -> dynamic)?\n}\n\n/**\n * Exposes the JavaScript
[NavigatorConcurrentHardware](https://developer.mozilla.org/en/docs/Web/API/NavigatorConcurrentHardware) to
Kotlin\n *\npublic external interface NavigatorConcurrentHardware {\n val hardwareConcurrency:
Number\n}\n\n/**\n * Exposes the JavaScript
[WorkerNavigator](https://developer.mozilla.org/en/docs/Web/API/WorkerNavigator) to Kotlin\n *\npublic
external abstract class WorkerNavigator : NavigatorID, NavigatorLanguage, NavigatorOnLine,
NavigatorConcurrentHardware {\n open val serviceWorker: ServiceWorkerContainer\n}\n\n/**\n * Exposes the
JavaScript [WorkerLocation](https://developer.mozilla.org/en/docs/Web/API/WorkerLocation) to Kotlin\n
*\npublic external abstract class WorkerLocation {\n open val href: String\n open val origin: String\n open val
protocol: String\n open val host: String\n open val hostname: String\n open val port: String\n open val
pathname: String\n open val search: String\n open val hash: String\n}\n\n/**\n * Exposes the JavaScript
[Storage](https://developer.mozilla.org/en/docs/Web/API/Storage) to Kotlin\n *\npublic external abstract class
Storage {\n open val length: Int\n fun key(index: Int): String?\n fun removeItem(key: String)\n fun clear()\n
fun getItem(key: String): String?\n fun setItem(key: String, value:
String)\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Storage.get(key: String):
String? = asDynamic()[key]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Storage.set(key: String, value:
String) { asDynamic()[key] = value }\n\n/**\n * Exposes the JavaScript
[WindowSessionStorage](https://developer.mozilla.org/en/docs/Web/API/WindowSessionStorage) to Kotlin\n
*\npublic external interface WindowSessionStorage {\n val sessionStorage: Storage\n}\n\n/**\n * Exposes the
JavaScript [WindowLocalStorage](https://developer.mozilla.org/en/docs/Web/API/WindowLocalStorage) to
Kotlin\n *\npublic external interface WindowLocalStorage {\n val localStorage: Storage\n}\n\n/**\n * Exposes
the JavaScript [StorageEvent](https://developer.mozilla.org/en/docs/Web/API/StorageEvent) to Kotlin\n
*\npublic external open class StorageEvent(type: String, eventInitDict: StorageEventInit = definedExternally) : Event {\n
open val key: String?\n open val oldValue: String?\n open val newValue: String?\n open val url: String\n
open val storageArea: Storage?\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE:

```

```

Short\n    val AT_TARGET: Short\n    val BUBBLING_PHASE: Short\n    }\n}\n\npublic external interface
StorageEventInit : EventInit {\n    var key: String? /* = null */\n        get() = definedExternally\n        set(value) =
definedExternally\n    var oldValue: String? /* = null */\n        get() = definedExternally\n        set(value) =
definedExternally\n    var newValue: String? /* = null */\n        get() = definedExternally\n        set(value) =
definedExternally\n    var url: String? /* = \"\" */\n        get() = definedExternally\n        set(value) =
definedExternally\n    var storageArea: Storage? /* = null */\n        get() = definedExternally\n        set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun StorageEventInit(key: String? = null,
oldValue: String? = null, newValue: String? = null, url: String? = \"\", storageArea: Storage? = null, bubbles:
Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): StorageEventInit {\n    val o =
js(\"({})\")\n    o[\"key\"] = key\n    o[\"oldValue\"] = oldValue\n    o[\"newValue\"] = newValue\n    o[\"url\"] =
url\n    o[\"storageArea\"] = storageArea\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return o\n}\n\npublic external abstract class HTMLAppletElement :
HTMLElement {\n    open var align: String\n    open var alt: String\n    open var archive: String\n    open var code:
String\n    open var codeBase: String\n    open var height: String\n    open var hspace: Int\n    open var name:
String\n    open var _object: String\n    open var vspace: Int\n    open var width: String\n\n    companion object {\n
        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLMarqueeElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMarqueeElement) to Kotlin\n
*/\n\npublic external abstract class HTMLMarqueeElement : HTMLElement {\n    open var behavior: String\n    open
var bgColor: String\n    open var direction: String\n    open var height: String\n    open var hspace: Int\n    open
var loop: Int\n    open var scrollAmount: Int\n    open var scrollDelay: Int\n    open var trueSpeed: Boolean\n    open
var vspace: Int\n    open var width: String\n    open var onbounce: ((Event) -> dynamic)?\n    open var onfinish: ((Event)
-> dynamic)?\n    open var onstart: ((Event) -> dynamic)?\n    fun start()\n    fun stop()\n\n    companion object {\n
        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLFrameSetElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFrameSetElement) to Kotlin\n
*/\n\npublic external abstract class HTMLFrameSetElement : HTMLElement, WindowEventHandlers {\n    open var
cols: String\n    open var rows: String\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val

```

```

DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external abstract class
HTMLFrameElement : HTMLElement {\n    open var name: String\n    open var scrolling: String\n    open var src:
String\n    open var frameBorder: String\n    open var longDesc: String\n    open var noResize: Boolean\n    open val
contentDocument: Document?\n    open val contentWindow: Window?\n    open var marginHeight: String\n    open
var marginWidth: String\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external abstract class
HTMLDirectoryElement : HTMLElement {\n    open var compact: Boolean\n\n    companion object {\n        val
ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLFontElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFontElement) to Kotlin\n *\npublic
external abstract class HTMLFontElement : HTMLElement {\n    open var color: String\n    open var face: String\n
open var size: String\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external interface External
{\n    fun AddSearchProvider()\n    fun IsSearchProviderInstalled()\n}\n\npublic external interface EventInit {\n
var bubbles: Boolean? /* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n    var
cancelable: Boolean? /* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n    var
composed: Boolean? /* = false */\n    get() = definedExternally\n    set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\npublic inline fun EventInit(bubbles: Boolean? = false,
cancelable: Boolean? = false, composed: Boolean? = false): EventInit {\n    val o = js(\"({})\")\n    o[\"bubbles\"] =
bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return o\n}\n\n/**\n * Exposes the
JavaScript [CustomEvent](https://developer.mozilla.org/en/docs/Web/API/CustomEvent) to Kotlin\n *\npublic

```

```

external open class CustomEvent(type: String, eventInitDict: CustomEventInit = definedExternally) : Event {
    open val detail: Any?
    fun initCustomEvent(type: String, bubbles: Boolean, cancelable: Boolean, detail: Any?)
    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
    public external interface CustomEventInit : EventInit {
        var detail: Any? /* = null */
        get() = definedExternally
        set(value) = definedExternally
    }
    @Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
    @kotlin.internal.InlineOnly
    public inline fun CustomEventInit(detail: Any? = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): CustomEventInit {
        val o = js("{}")
        o["detail"] = detail
        o["bubbles"] = bubbles
        o["cancelable"] = cancelable
        o["composed"] = composed
        return o
    }
    public external interface EventListenerOptions {
        var capture: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    }
    @Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
    @kotlin.internal.InlineOnly
    public inline fun EventListenerOptions(capture: Boolean? = false): EventListenerOptions {
        val o = js("{}")
        o["capture"] = capture
        return o
    }
    public external interface AddEventListenerOptions : EventListenerOptions {
        var passive: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
        var once: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    }
    @Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
    @kotlin.internal.InlineOnly
    public inline fun AddEventListenerOptions(passive: Boolean? = false, once: Boolean? = false, capture: Boolean? = false): AddEventListenerOptions {
        val o = js("{}")
        o["passive"] = passive
        o["once"] = once
        o["capture"] = capture
        return o
    }
    public external interface NonElementParentNode {
        fun getElementById(elementId: String): Element?
    }
    /**
     * Exposes the JavaScript [DocumentOrShadowRoot](https://developer.mozilla.org/en/docs/Web/API/DocumentOrShadowRoot) to Kotlin
     */
    public external interface DocumentOrShadowRoot {
        val fullscreenElement: Element?
        get() = definedExternally
    }
    /**
     * Exposes the JavaScript [ParentNode](https://developer.mozilla.org/en/docs/Web/API/ParentNode) to Kotlin
     */
    public external interface ParentNode {
        val children: HTMLCollection
        val firstElementChild: Element?
        get() = definedExternally
        val lastElementChild: Element?
        get() = definedExternally
        val childElementCount: Int
        fun prepend(vararg nodes: dynamic)
        fun append(vararg nodes: dynamic)
        fun querySelector(selectors: String): Element?
        fun querySelectorAll(selectors: String): NodeList
    }
    /**
     * Exposes the JavaScript [NonDocumentTypeChildNode](https://developer.mozilla.org/en/docs/Web/API/NonDocumentTypeChildNode) to Kotlin
     */
    public external interface NonDocumentTypeChildNode {
        val previousElementSibling: Element?
        get() = definedExternally
        val nextElementSibling: Element?
        get() = definedExternally
    }
    /**
     * Exposes the JavaScript [ChildNode](https://developer.mozilla.org/en/docs/Web/API/ChildNode) to Kotlin
     */
    public external interface ChildNode {
        fun before(vararg nodes: dynamic)
        fun after(vararg nodes: dynamic)
        fun replaceWith(vararg nodes: dynamic)
        fun remove()
    }
    /**
     * Exposes the JavaScript [Slotable](https://developer.mozilla.org/en/docs/Web/API/Slotable) to Kotlin
     */
    public external interface Slotable {
        val assignedSlot: HTMLSlotElement?
        get() = definedExternally
    }
    /**
     * Exposes the JavaScript [NodeList](https://developer.mozilla.org/en/docs/Web/API/NodeList) to Kotlin
     */
    public external abstract class NodeList : ItemArrayLike<Node> {
        override fun item(index: Int): Node?
    }
    @Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
    @kotlin.internal.InlineOnly
    public inline operator fun NodeList.get(index: Int): Node? = asDynamic()[index]
    /**
     * Exposes the JavaScript [HTMLCollection](https://developer.mozilla.org/en/docs/Web/API/HTMLCollection) to Kotlin
     */
    public external abstract class HTMLCollection : ItemArrayLike<Element>, UnionElementOrHTMLCollection {
        override fun item(index: Int): Element?
        fun namedItem(name: String): Element?
    }
    @Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
    @kotlin.internal.InlineOnly
    public inline operator fun HTMLCollection.get(index:

```

```

Int): Element? = asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun HTMLCollection.get(name:
String): Element? = asDynamic()[name]\n\n/**\n * Exposes the JavaScript
[MutationObserver](https://developer.mozilla.org/en/docs/Web/API/MutationObserver) to Kotlin\n */\npublic
external open class MutationObserver(callback: (Array<MutationRecord>, MutationObserver) -> Unit) {\n fun
observe(target: Node, options: MutationObserverInit = definedExternally)\n fun disconnect()\n fun
takeRecords(): Array<MutationRecord>\n}\n\n/**\n * Exposes the JavaScript
[MutationObserverInit](https://developer.mozilla.org/en/docs/Web/API/MutationObserverInit) to Kotlin\n
*/\npublic external interface MutationObserverInit {\n var childList: Boolean? /* = false */\n get() =
definedExternally\n set(value) = definedExternally\n var attributes: Boolean?\n get() =
definedExternally\n set(value) = definedExternally\n var characterData: Boolean?\n get() =
definedExternally\n set(value) = definedExternally\n var subtree: Boolean? /* = false */\n get() =
definedExternally\n set(value) = definedExternally\n var attributeOldValue: Boolean?\n get() =
definedExternally\n set(value) = definedExternally\n var characterDataOldValue: Boolean?\n get() =
definedExternally\n set(value) = definedExternally\n var attributeFilter: Array<String>?\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun MutationObserverInit(childList:
Boolean? = false, attributes: Boolean? = undefined, characterData: Boolean? = undefined, subtree: Boolean? = false,
attributeOldValue: Boolean? = undefined, characterDataOldValue: Boolean? = undefined, attributeFilter:
Array<String>? = undefined): MutationObserverInit {\n val o = js("{}")\n o["childList"] = childList\n
o["attributes"] = attributes\n o["characterData"] = characterData\n o["subtree"] = subtree\n
o["attributeOldValue"] = attributeOldValue\n o["characterDataOldValue"] = characterDataOldValue\n
o["attributeFilter"] = attributeFilter\n return o\n}\n\n/**\n * Exposes the JavaScript
[MutationRecord](https://developer.mozilla.org/en/docs/Web/API/MutationRecord) to Kotlin\n */\npublic external
abstract class MutationRecord {\n open val type: String\n open val target: Node\n open val addedNodes:
NodeList\n open val removedNodes: NodeList\n open val previousSibling: Node?\n open val nextSibling:
Node?\n open val attributeName: String?\n open val attributeNamespace: String?\n open val oldValue:
String?\n}\n\n/**\n * Exposes the JavaScript [Node](https://developer.mozilla.org/en/docs/Web/API/Node) to
Kotlin\n */\npublic external abstract class Node : EventTarget {\n open val nodeType: Short\n open val
nodeName: String\n open val baseURI: String\n open val isConnected: Boolean\n open val ownerDocument:
Document?\n open val parentNode: Node?\n open val parentElement: Element?\n open val childNodes:
NodeList\n open val firstChild: Node?\n open val lastChild: Node?\n open val previousSibling: Node?\n
open val nextSibling: Node?\n open var nodeValue: String?\n open var textContent: String?\n fun
getRootNode(options: GetRootNodeOptions = definedExternally): Node\n fun hasChildNodes(): Boolean\n fun
normalize()\n fun cloneNode(deep: Boolean = definedExternally): Node\n fun isEqualNode(otherNode: Node?):
Boolean\n fun isSameNode(otherNode: Node?): Boolean\n fun compareDocumentPosition(other: Node): Short\n
fun contains(other: Node?): Boolean\n fun lookupPrefix(namespace: String?): String?\n fun
lookupNamespaceURI(prefix: String?): String?\n fun isDefaultNamespace(namespace: String?): Boolean\n fun
insertBefore(node: Node, child: Node?): Node\n fun appendChild(node: Node): Node\n fun replaceChild(node:
Node, child: Node): Node\n fun removeChild(child: Node): Node\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

```

```

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n\n\npublic external interface
GetRootNodeOptions {\n var composed: Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun GetRootNodeOptions(composed:
Boolean? = false): GetRootNodeOptions {\n val o = js(\"({})\")\n o[\"composed\"] = composed\n return
o\n}\n\n/**\n * Exposes the JavaScript [Document](https://developer.mozilla.org/en/docs/Web/API/Document) to
Kotlin\n */\npublic external open class Document : Node, GlobalEventHandlers,
DocumentAndElementEventHandlers, NonElementParentNode, DocumentOrShadowRoot, ParentNode,
GeometryUtils {\n open val implementation: DOMImplementation\n open val URL: String\n open val
documentURI: String\n open val origin: String\n open val compatMode: String\n open val characterSet:
String\n open val charset: String\n open val inputEncoding: String\n open val contentType: String\n open val
doctype: DocumentType?\n open val documentElement: Element?\n open val location: Location?\n var
domain: String\n open val referrer: String\n var cookie: String\n open val lastModified: String\n open val
readyState: DocumentReadyState\n var title: String\n var dir: String\n var body: HTMLElement?\n open val
head: HTMLHeadElement?\n open val images: HTMLCollection\n open val embeds: HTMLCollection\n open
val plugins: HTMLCollection\n open val links: HTMLCollection\n open val forms: HTMLCollection\n open
val scripts: HTMLCollection\n open val currentScript: HTMLScriptElement?\n open val defaultView:
Window?\n open val activeElement: Element?\n var designMode: String\n var onreadystatechange: ((Event) ->
dynamic)?\n var fgColor: String\n var linkColor: String\n var vlinkColor: String\n var alinkColor: String\n
var bgColor: String\n open val anchors: HTMLCollection\n open val applets: HTMLCollection\n open val all:
HTMLAllCollection\n open val scrollingElement: Element?\n open val styleSheets: StyleSheetList\n open val
rootElement: SVGSVGElement?\n open val fullscreenEnabled: Boolean\n open val fullscreen: Boolean\n var
onfullscreenchange: ((Event) -> dynamic)?\n var onfullscreenerror: ((Event) -> dynamic)?\n override var
onabort: ((Event) -> dynamic)?\n override var onblur: ((FocusEvent) -> dynamic)?\n override var oncancel:
((Event) -> dynamic)?\n override var oncanplay: ((Event) -> dynamic)?\n override var oncanplaythrough:
((Event) -> dynamic)?\n override var onchange: ((Event) -> dynamic)?\n override var onclick: ((MouseEvent) ->
dynamic)?\n override var onclose: ((Event) -> dynamic)?\n override var oncontextmenu: ((MouseEvent) ->
dynamic)?\n override var oncuechange: ((Event) -> dynamic)?\n override var ondblclick: ((MouseEvent) ->
dynamic)?\n override var ondrag: ((DragEvent) -> dynamic)?\n override var ondragend: ((DragEvent) ->
dynamic)?\n override var ondragenter: ((DragEvent) -> dynamic)?\n override var ondragexit: ((DragEvent) ->
dynamic)?\n override var ondragleave: ((DragEvent) -> dynamic)?\n override var ondragover: ((DragEvent) ->
dynamic)?\n override var ondragstart: ((DragEvent) -> dynamic)?\n override var ondrop: ((DragEvent) ->
dynamic)?\n override var ondurationchange: ((Event) -> dynamic)?\n override var onemptied: ((Event) ->
dynamic)?\n override var onended: ((Event) -> dynamic)?\n override var onerror: ((dynamic, String, Int, Int,
Any?) -> dynamic)?\n override var onfocus: ((FocusEvent) -> dynamic)?\n override var oninput: ((InputEvent) ->
dynamic)?\n override var oninvalid: ((Event) -> dynamic)?\n override var onkeydown: ((KeyboardEvent) ->
dynamic)?\n override var onkeypress: ((KeyboardEvent) -> dynamic)?\n override var onkeyup:
((KeyboardEvent) -> dynamic)?\n override var onload: ((Event) -> dynamic)?\n override var onloadeddata:
((Event) -> dynamic)?\n override var onloadedmetadata: ((Event) -> dynamic)?\n override var onloadend:
((Event) -> dynamic)?\n override var onloadstart: ((ProgressEvent) -> dynamic)?\n override var onmousedown:
((MouseEvent) -> dynamic)?\n override var onmouseenter: ((MouseEvent) -> dynamic)?\n override var
onmouseleave: ((MouseEvent) -> dynamic)?\n override var onmousemove: ((MouseEvent) -> dynamic)?\n
override var onmouseout: ((MouseEvent) -> dynamic)?\n override var onmouseover: ((MouseEvent) ->
dynamic)?\n override var onmouseup: ((MouseEvent) -> dynamic)?\n override var onwheel: ((WheelEvent) ->
dynamic)?\n override var onpause: ((Event) -> dynamic)?\n override var onplay: ((Event) -> dynamic)?\n
override var onplaying: ((Event) -> dynamic)?\n override var onprogress: ((ProgressEvent) -> dynamic)?\n
override var onratechange: ((Event) -> dynamic)?\n override var onreset: ((Event) -> dynamic)?\n override var
onresize: ((Event) -> dynamic)?\n override var onscroll: ((Event) -> dynamic)?\n override var onseeked:

```



```

((Event) -> dynamic)?\n  override var onseeking: ((Event) -> dynamic)?\n  override var onselect: ((Event) ->
dynamic)?\n  override var onshow: ((Event) -> dynamic)?\n  override var onstalled: ((Event) -> dynamic)?\n
override var onsubmit: ((Event) -> dynamic)?\n  override var onsuspend: ((Event) -> dynamic)?\n  override var
ontimeupdate: ((Event) -> dynamic)?\n  override var ontoggle: ((Event) -> dynamic)?\n  override var
onvolumechange: ((Event) -> dynamic)?\n  override var onwaiting: ((Event) -> dynamic)?\n  override var
ongotpointercapture: ((PointerEvent) -> dynamic)?\n  override var onlostpointercapture: ((PointerEvent) ->
dynamic)?\n  override var onpointerdown: ((PointerEvent) -> dynamic)?\n  override var onpointermove:
((PointerEvent) -> dynamic)?\n  override var onpointerup: ((PointerEvent) -> dynamic)?\n  override var
onpointercancel: ((PointerEvent) -> dynamic)?\n  override var onpointerover: ((PointerEvent) -> dynamic)?\n
override var onpointerout: ((PointerEvent) -> dynamic)?\n  override var onpointerenter: ((PointerEvent) ->
dynamic)?\n  override var onpointerleave: ((PointerEvent) -> dynamic)?\n  override var oncopy:
((ClipboardEvent) -> dynamic)?\n  override var oncut: ((ClipboardEvent) -> dynamic)?\n  override var onpaste:
((ClipboardEvent) -> dynamic)?\n  override val fullscreenElement: Element?\n  override val children:
HTMLCollection\n  override val firstElementChild: Element?\n  override val lastElementChild: Element?\n
override val childElementCount: Int\n  fun getElementsByTagName(qualifiedName: String): HTMLCollection\n
fun getElementsByTagNameNS(namespace: String?, localName: String): HTMLCollection\n  fun
getElementsByTagName(className: String): HTMLCollection\n  fun createElement(localName: String,
options: ElementCreationOptions = definedExternally): Element\n  fun createElementNS(namespace: String?,
qualifiedName: String, options: ElementCreationOptions = definedExternally): Element\n  fun
createDocumentFragment(): DocumentFragment\n  fun createTextNode(data: String): Text\n  fun
createCDATASection(data: String): CDATASection\n  fun createComment(data: String): Comment\n  fun
createProcessingInstruction(target: String, data: String): ProcessingInstruction\n  fun importNode(node: Node,
deep: Boolean = definedExternally): Node\n  fun adoptNode(node: Node): Node\n  fun
createAttribute(localName: String): Attr\n  fun createAttributeNS(namespace: String?, qualifiedName: String):
Attr\n  fun createEvent(`interface`: String): Event\n  fun createRange(): Range\n  fun createNodeIterator(root:
Node, whatToShow: Int = definedExternally, filter: NodeFilter? = definedExternally): NodeIterator\n  fun
createNodeIterator(root: Node, whatToShow: Int = definedExternally, filter: ((Node) -> Short)? =
definedExternally): NodeIterator\n  fun createTreeWalker(root: Node, whatToShow: Int = definedExternally, filter:
NodeFilter? = definedExternally): TreeWalker\n  fun createTreeWalker(root: Node, whatToShow: Int =
definedExternally, filter: ((Node) -> Short)? = definedExternally): TreeWalker\n  fun
getElementsByTagName(elementName: String): NodeList\n  fun open(type: String = definedExternally, replace:
String = definedExternally): Document\n  fun open(url: String, name: String, features: String): Window\n  fun
close()\n  fun write(vararg text: String)\n  fun writeln(vararg text: String)\n  fun hasFocus(): Boolean\n  fun
execCommand(commandId: String, showUI: Boolean = definedExternally, value: String = definedExternally):
Boolean\n  fun queryCommandEnabled(commandId: String): Boolean\n  fun
queryCommandIndeterm(commandId: String): Boolean\n  fun queryCommandState(commandId: String):
Boolean\n  fun queryCommandSupported(commandId: String): Boolean\n  fun
queryCommandValue(commandId: String): String\n  fun clear()\n  fun captureEvents()\n  fun releaseEvents()\n
fun elementFromPoint(x: Double, y: Double): Element?\n  fun elementsFromPoint(x: Double, y: Double):
Array<Element>\n  fun caretPositionFromPoint(x: Double, y: Double): CaretPosition?\n  fun createTouch(view:
Window, target: EventTarget, identifier: Int, pageX: Int, pageY: Int, screenX: Int, screenY: Int): Touch\n  fun
createTouchList(vararg touches: Touch): TouchList\n  fun exitFullscreen(): Promise<Unit>\n  override fun
getElementById(elementId: String): Element?\n  override fun prepend(vararg nodes: dynamic)\n  override fun
append(vararg nodes: dynamic)\n  override fun querySelector(selectors: String): Element?\n  override fun
querySelectorAll(selectors: String): NodeList\n  override fun getBoxQuads(options: BoxQuadOptions /* =
definedExternally */): Array<DOMQuad>\n  override fun convertQuadFromNode(quad: dynamic, from: dynamic,
options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n  override fun
convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* =

```

```

definedExternally */): DOMQuad\n    override fun convertPointFromNode(point: DOMPointInit, from: dynamic,
options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint\n\n    companion object {\n        val
ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n
    }\n\n    @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n    @kotlin.internal.InlineOnly\n    public inline operator fun Document.get(name: String):
dynamic = asDynamic()[name]\n\n    /**\n     * Exposes the JavaScript
[XMLDocument](https://developer.mozilla.org/en/docs/Web/API/XMLDocument) to Kotlin\n     *\n     public external
open class XMLDocument : Document {\n        companion object {\n            val ELEMENT_NODE: Short\n            val
ATTRIBUTE_NODE: Short\n            val TEXT_NODE: Short\n            val CDATA_SECTION_NODE: Short\n            val
ENTITY_REFERENCE_NODE: Short\n            val ENTITY_NODE: Short\n            val
PROCESSING_INSTRUCTION_NODE: Short\n            val COMMENT_NODE: Short\n            val
DOCUMENT_NODE: Short\n            val DOCUMENT_TYPE_NODE: Short\n            val
DOCUMENT_FRAGMENT_NODE: Short\n            val NOTATION_NODE: Short\n            val
DOCUMENT_POSITION_DISCONNECTED: Short\n            val DOCUMENT_POSITION_PRECEDING: Short\n
            val DOCUMENT_POSITION_FOLLOWING: Short\n            val DOCUMENT_POSITION_CONTAINS: Short\n
            val DOCUMENT_POSITION_CONTAINED_BY: Short\n            val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n        }\n\n        public external interface
ElementCreationOptions {\n            var `is`: String?\n            get() = definedExternally\n            set(value) =
definedExternally\n        }\n\n        @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n        @kotlin.internal.InlineOnly\n        public inline fun ElementCreationOptions(`is`: String?
= undefined): ElementCreationOptions {\n            val o = js("{}")\n            o["is"] = `is`\n            return o\n        }\n\n        /**\n         * Exposes the JavaScript
[DOMImplementation](https://developer.mozilla.org/en/docs/Web/API/DOMImplementation) to Kotlin\n         *\n         public external abstract class DOMImplementation {\n            fun createDocumentType(qualifiedName: String,
publicId: String, systemId: String): DocumentType\n            fun createDocument(namespace: String?, qualifiedName:
String, doctype: DocumentType? = definedExternally): XMLDocument\n            fun createHTMLDocument(title: String
= definedExternally): Document\n            fun hasFeature(): Boolean\n        }\n\n        /**\n         * Exposes the JavaScript
[DocumentType](https://developer.mozilla.org/en/docs/Web/API/DocumentType) to Kotlin\n         *\n         public external
abstract class DocumentType : Node, ChildNode {\n            open val name: String\n            open val publicId: String\n            open
val systemId: String\n\n            companion object {\n                val ELEMENT_NODE: Short\n                val ATTRIBUTE_NODE:
Short\n                val TEXT_NODE: Short\n                val CDATA_SECTION_NODE: Short\n                val
ENTITY_REFERENCE_NODE: Short\n                val ENTITY_NODE: Short\n                val
PROCESSING_INSTRUCTION_NODE: Short\n                val COMMENT_NODE: Short\n                val
DOCUMENT_NODE: Short\n                val DOCUMENT_TYPE_NODE: Short\n                val
DOCUMENT_FRAGMENT_NODE: Short\n                val NOTATION_NODE: Short\n                val
DOCUMENT_POSITION_DISCONNECTED: Short\n                val DOCUMENT_POSITION_PRECEDING: Short\n
                val DOCUMENT_POSITION_FOLLOWING: Short\n                val DOCUMENT_POSITION_CONTAINS: Short\n
                val DOCUMENT_POSITION_CONTAINED_BY: Short\n                val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n            }\n\n            /**\n             * Exposes the JavaScript
[DocumentFragment](https://developer.mozilla.org/en/docs/Web/API/DocumentFragment) to Kotlin\n             *\n             public

```

```

external open class DocumentFragment : Node, NonElementParentNode, ParentNode {
    override val children: HTMLCollection
    override val firstElementChild: Element?
    override val lastElementChild: Element?
    override val childElementCount: Int
    override fun getElementById(elementId: String): Element?
    override fun prepend(vararg nodes: dynamic)
    override fun append(vararg nodes: dynamic)
    override fun querySelector(selectors: String): Element?
    override fun querySelectorAll(selectors: String): NodeList
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/* Exposes the JavaScript [ShadowRoot](https://developer.mozilla.org/en/docs/Web/API/ShadowRoot) to Kotlin */
@public external open class ShadowRoot : DocumentFragment, DocumentOrShadowRoot {
    open val mode: ShadowRootMode
    open val host: Element
    override val fullscreenElement: Element?
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/* Exposes the JavaScript [Element](https://developer.mozilla.org/en/docs/Web/API/Element) to Kotlin */
@public external abstract class Element : Node, ParentNode, NonDocumentTypeChildNode, ChildNode, Slotable, GeometryUtils, UnionElementOrHTMLCollection, UnionElementOrRadioNodeList, UnionElementOrMouseEvent, UnionElementOrProcessingInstruction {
    open val namespaceURI: String?
    open val prefix: String?
    open val localName: String
    open val tagName: String
    open var id: String
    open var className: String
    open val classList: DOMTokenList
    open var slot: String
    open val attributes: NamedNodeMap
    open val shadowRoot: ShadowRoot?
    open var scrollTop: Double
    open var scrollLeft: Double
    open val scrollWidth: Int
    open val scrollHeight: Int
    open val clientTop: Int
    open val clientLeft: Int
    open val clientWidth: Int
    open val clientHeight: Int
    open var innerHTML: String
    open var outerHTML: String
    fun hasAttributes(): Boolean
    fun getAttributeNames(): Array<String>
    fun getAttribute(qualifiedName: String): String?
    fun getAttributeNS(namespace: String?, localName: String): String?
    fun setAttribute(qualifiedName: String, value: String)
    fun setAttributeNS(namespace: String?, qualifiedName: String, value: String)
    fun removeAttribute(qualifiedName: String)
    fun removeAttributeNS(namespace: String?, localName: String)
    fun hasAttribute(qualifiedName: String): Boolean
    fun hasAttributeNS(namespace: String?, localName: String): Boolean
    fun getAttributeNode(qualifiedName: String): Attr?
    fun getAttributeNodeNS(namespace: String?, localName: String): Attr?
    fun setAttributeNode(attr: Attr): Attr?
    fun setAttributeNodeNS(attr: Attr): Attr?
    fun removeAttributeNode(attr: Attr): Attr
    fun attachShadow(init: ShadowRootInit): ShadowRoot
    fun closest(selectors: String): Element?
    fun matches(selectors: String): Boolean
    fun webkitMatchesSelector(selectors: String): Boolean
    fun getElementsByTagName(qualifiedName: String): HTMLCollection
    fun getElementsByTagNameNS(namespace: String?, localName: String): HTMLCollection
    fun getElementsByTagNameNS(classNames: String): HTMLCollection
    fun insertAdjacentElement(where: String,

```



```

ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }|}|}|n/n/**\n * Exposes the JavaScript
[Text](https://developer.mozilla.org/en/docs/Web/API/Text) to Kotlin\n */\npublic external open class Text(data:
String = definedExternally) : CharacterData, Slotable, GeometryUtils {\n    open val wholeText: String\n    override
val assignedSlot: HTMLSlotElement?\n    override val previousElementSibling: Element?\n    override val
nextElementSibling: Element?\n    fun splitText(offset: Int): Text\n    override fun getBoxQuads(options:
BoxQuadOptions /* = definedExternally */): Array<DOMQuad>\n    override fun convertQuadFromNode(quad:
dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n    override
fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* =
definedExternally */): DOMQuad\n    override fun convertPointFromNode(point: DOMPointInit, from: dynamic,
options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint\n    override fun before(vararg nodes:
dynamic)\n    override fun after(vararg nodes: dynamic)\n    override fun replaceWith(vararg nodes: dynamic)\n
override fun remove()\n\n    companion object {\n        val ELEMENT_NODE: Short\n    val
ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val
ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }|}|}|n/n/**\n * Exposes the JavaScript
[CDATASection](https://developer.mozilla.org/en/docs/Web/API/CDATASection) to Kotlin\n */\npublic external
open class CDATASection : Text {\n    companion object {\n        val ELEMENT_NODE: Short\n    val
ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val
ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }|}|}|n/n/**\n * Exposes the JavaScript
[ProcessingInstruction](https://developer.mozilla.org/en/docs/Web/API/ProcessingInstruction) to Kotlin\n
*/\npublic external abstract class ProcessingInstruction : CharacterData, LinkStyle,
UnionElementOrProcessingInstruction {\n    open val target: String\n\n    companion object {\n        val
ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val
CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE:
Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n

```

```

    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[Comment](https://developer.mozilla.org/en/docs/Web/API/Comment) to Kotlin\n\n *\n\npublic external open class
Comment(data: String = definedExternally) : CharacterData {\n    override val previousElementSibling: Element?\n
    override val nextElementSibling: Element?\n    override fun before(vararg nodes: dynamic)\n    override fun
after(vararg nodes: dynamic)\n    override fun replaceWith(vararg nodes: dynamic)\n    override fun remove()\n\n
companion object {\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val
TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE:
Short\n    val ENTITY_NODE: Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val
COMMENT_NODE: Short\n    val DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n
    val DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[Range](https://developer.mozilla.org/en/docs/Web/API/Range) to Kotlin\n\n *\n\npublic external open class Range {\n
    open val startContainer: Node\n    open val startOffset: Int\n    open val endContainer: Node\n    open val
endOffset: Int\n    open val collapsed: Boolean\n    open val commonAncestorContainer: Node\n    fun setStart(node:
Node, offset: Int)\n    fun setEnd(node: Node, offset: Int)\n    fun setStartBefore(node: Node)\n    fun
setStartAfter(node: Node)\n    fun setEndBefore(node: Node)\n    fun setEndAfter(node: Node)\n    fun
collapse(toStart: Boolean = definedExternally)\n    fun selectNode(node: Node)\n    fun selectNodeContents(node:
Node)\n    fun compareBoundaryPoints(how: Short, sourceRange: Range): Short\n    fun deleteContents()\n    fun
extractContents(): DocumentFragment\n    fun cloneContents(): DocumentFragment\n    fun insertNode(node:
Node)\n    fun surroundContents(newParent: Node)\n    fun cloneRange(): Range\n    fun detach()\n    fun
isPointInRange(node: Node, offset: Int): Boolean\n    fun comparePoint(node: Node, offset: Int): Short\n    fun
intersectsNode(node: Node): Boolean\n    fun getClientRects(): Array<DOMRect>\n    fun
getBoundingClientRect(): DOMRect\n    fun createContextualFragment(fragment: String): DocumentFragment\n\n
companion object {\n    val START_TO_START: Short\n    val START_TO_END: Short\n    val
END_TO_END: Short\n    val END_TO_START: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[NodeIterator](https://developer.mozilla.org/en/docs/Web/API/NodeIterator) to Kotlin\n\n *\n\npublic external abstract
class NodeIterator {\n    open val root: Node\n    open val referenceNode: Node\n    open val
pointerBeforeReferenceNode: Boolean\n    open val whatToShow: Int\n    open val filter: NodeFilter?\n    fun
nextNode(): Node?\n    fun previousNode(): Node?\n    fun detach()\n}\n\n/**\n * Exposes the JavaScript
[TreeWalker](https://developer.mozilla.org/en/docs/Web/API/TreeWalker) to Kotlin\n\n *\n\npublic external abstract
class TreeWalker {\n    open val root: Node\n    open val whatToShow: Int\n    open val filter: NodeFilter?\n    open
var currentNode: Node\n    fun parentNode(): Node?\n    fun firstChild(): Node?\n    fun lastChild(): Node?\n    fun
previousSibling(): Node?\n    fun nextSibling(): Node?\n    fun previousNode(): Node?\n    fun nextNode():
Node?\n}\n\n/**\n * Exposes the JavaScript
[NodeFilter](https://developer.mozilla.org/en/docs/Web/API/NodeFilter) to Kotlin\n\n
*\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external interface NodeFilter {\n
    fun acceptNode(node: Node): Short\n\n    companion object {\n        val FILTER_ACCEPT: Short\n        val
FILTER_REJECT: Short\n        val FILTER_SKIP: Short\n        val SHOW_ALL: Int\n        val
SHOW_ELEMENT: Int\n        val SHOW_ATTRIBUTE: Int\n        val SHOW_TEXT: Int\n        val
SHOW_CDATA_SECTION: Int\n        val SHOW_ENTITY_REFERENCE: Int\n        val SHOW_ENTITY: Int\n
        val SHOW_PROCESSING_INSTRUCTION: Int\n        val SHOW_COMMENT: Int\n        val
SHOW_DOCUMENT: Int\n        val SHOW_DOCUMENT_TYPE: Int\n        val
SHOW_DOCUMENT_FRAGMENT: Int\n        val SHOW_NOTATION: Int\n    }\n}\n\n/**\n * Exposes the

```

JavaScript [DOMTokenList](https://developer.mozilla.org/en/docs/Web/API/DOMTokenList) to Kotlin\n */\npublic external abstract class DOMTokenList : ItemArrayLike<String> {\n open var value: String\n fun contains(token: String): Boolean\n fun add(vararg tokens: String)\n fun remove(vararg tokens: String)\n fun toggle(token: String, force: Boolean = definedExternally): Boolean\n fun replace(token: String, newToken: String)\n fun supports(token: String): Boolean\n override fun item(index: Int): String?\n}\n\n@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun DOMTokenList.get(index: Int): String? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript [DOMPointReadOnly](https://developer.mozilla.org/en/docs/Web/API/DOMPointReadOnly) to Kotlin\n */\npublic external open class DOMPointReadOnly(x: Double, y: Double, z: Double, w: Double) {\n open val x: Double\n open val y: Double\n open val z: Double\n open val w: Double\n fun matrixTransform(matrix: DOMMatrixReadOnly): DOMPoint}\n\n/**\n * Exposes the JavaScript [DOMPoint](https://developer.mozilla.org/en/docs/Web/API/DOMPoint) to Kotlin\n */\npublic external open class DOMPoint : DOMPointReadOnly {\n constructor(point: DOMPointInit)\n constructor(x: Double = definedExternally, y: Double = definedExternally, z: Double = definedExternally, w: Double = definedExternally)\n override var x: Double\n override var y: Double\n override var z: Double\n override var w: Double}\n\n/**\n * Exposes the JavaScript [DOMPointInit](https://developer.mozilla.org/en/docs/Web/API/DOMPointInit) to Kotlin\n */\npublic external interface DOMPointInit {\n var x: Double? /* = 0.0 */\n get() = definedExternally\n set(value) = definedExternally\n var y: Double? /* = 0.0 */\n get() = definedExternally\n set(value) = definedExternally\n var z: Double? /* = 0.0 */\n get() = definedExternally\n set(value) = definedExternally\n var w: Double? /* = 1.0 */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun DOMPointInit(x: Double? = 0.0, y: Double? = 0.0, z: Double? = 0.0, w: Double? = 1.0): DOMPointInit {\n val o = js("{}")\n o["x"] = x\n o["y"] = y\n o["z"] = z\n o["w"] = w\n return o}\n\n/**\n * Exposes the JavaScript [DOMRect](https://developer.mozilla.org/en/docs/Web/API/DOMRect) to Kotlin\n */\npublic external open class DOMRect(x: Double = definedExternally, y: Double = definedExternally, width: Double = definedExternally, height: Double = definedExternally) : DOMRectReadOnly {\n override var x: Double\n override var y: Double\n override var width: Double\n override var height: Double}\n\n/**\n * Exposes the JavaScript [DOMRectReadOnly](https://developer.mozilla.org/en/docs/Web/API/DOMRectReadOnly) to Kotlin\n */\npublic external open class DOMRectReadOnly(x: Double, y: Double, width: Double, height: Double) {\n open val x: Double\n open val y: Double\n open val width: Double\n open val height: Double\n open val top: Double\n open val right: Double\n open val bottom: Double\n open val left: Double}\n\npublic external interface DOMRectInit {\n var x: Double? /* = 0.0 */\n get() = definedExternally\n set(value) = definedExternally\n var y: Double? /* = 0.0 */\n get() = definedExternally\n set(value) = definedExternally\n var width: Double? /* = 0.0 */\n get() = definedExternally\n set(value) = definedExternally\n var height: Double? /* = 0.0 */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun DOMRectInit(x: Double? = 0.0, y: Double? = 0.0, width: Double? = 0.0, height: Double? = 0.0): DOMRectInit {\n val o = js("{}")\n o["x"] = x\n o["y"] = y\n o["width"] = width\n o["height"] = height\n return o}\n\npublic external interface DOMRectList : ItemArrayLike<DOMRect> {\n override fun item(index: Int): DOMRect?\n}\n\n@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun DOMRectList.get(index: Int): DOMRect? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript [DOMQuad](https://developer.mozilla.org/en/docs/Web/API/DOMQuad) to Kotlin\n */\npublic external open class DOMQuad {\n constructor(p1: DOMPointInit = definedExternally, p2: DOMPointInit = definedExternally, p3:

```

DOMPointInit = definedExternally, p4: DOMPointInit = definedExternally)\n  constructor(rect: DOMRectInit)\n
open val p1: DOMPoint\n  open val p2: DOMPoint\n  open val p3: DOMPoint\n  open val p4: DOMPoint\n
open val bounds: DOMRectReadOnly\n}\n\n/**\n * Exposes the JavaScript
[DOMMatrixReadOnly](https://developer.mozilla.org/en/docs/Web/API/DOMMatrixReadOnly) to Kotlin\n
*/\npublic external open class DOMMatrixReadOnly(numberSequence: Array<Double>) {\n  open val a: Double\n
  open val b: Double\n  open val c: Double\n  open val d: Double\n  open val e: Double\n  open val f: Double\n
  open val m11: Double\n  open val m12: Double\n  open val m13: Double\n  open val m14: Double\n  open val
  m21: Double\n  open val m22: Double\n  open val m23: Double\n  open val m24: Double\n  open val m31:
  Double\n  open val m32: Double\n  open val m33: Double\n  open val m34: Double\n  open val m41: Double\n
  open val m42: Double\n  open val m43: Double\n  open val m44: Double\n  open val is2D: Boolean\n  open
  val isIdentity: Boolean\n  fun translate(tx: Double, ty: Double, tz: Double = definedExternally): DOMMatrix\n
  fun scale(scale: Double, originX: Double = definedExternally, originY: Double = definedExternally): DOMMatrix\n
  fun scale3d(scale: Double, originX: Double = definedExternally, originY: Double = definedExternally, originZ:
  Double = definedExternally): DOMMatrix\n  fun scaleNonUniform(scaleX: Double, scaleY: Double =
  definedExternally, scaleZ: Double = definedExternally, originX: Double = definedExternally, originY: Double =
  definedExternally, originZ: Double = definedExternally): DOMMatrix\n  fun rotate(angle: Double, originX:
  Double = definedExternally, originY: Double = definedExternally): DOMMatrix\n  fun rotateFromVector(x:
  Double, y: Double): DOMMatrix\n  fun rotateAxisAngle(x: Double, y: Double, z: Double, angle: Double):
  DOMMatrix\n  fun skewX(sx: Double): DOMMatrix\n  fun skewY(sy: Double): DOMMatrix\n  fun
  multiply(other: DOMMatrix): DOMMatrix\n  fun flipX(): DOMMatrix\n  fun flipY(): DOMMatrix\n  fun
  inverse(): DOMMatrix\n  fun transformPoint(point: DOMPointInit = definedExternally): DOMPoint\n  fun
  toFloat32Array(): Float32Array\n  fun toFloat64Array(): Float64Array\n}\n\n/**\n * Exposes the JavaScript
[DOMMatrix](https://developer.mozilla.org/en/docs/Web/API/DOMMatrix) to Kotlin\n
*/\npublic external open
class DOMMatrix() : DOMMatrixReadOnly {\n  constructor(transformList: String)\n  constructor(other:
  DOMMatrixReadOnly)\n  constructor(array32: Float32Array)\n  constructor(array64: Float64Array)\n
  constructor(numberSequence: Array<Double>)\n  override var a: Double\n  override var b: Double\n  override
  var c: Double\n  override var d: Double\n  override var e: Double\n  override var f: Double\n  override var m11:
  Double\n  override var m12: Double\n  override var m13: Double\n  override var m14: Double\n  override var
  m21: Double\n  override var m22: Double\n  override var m23: Double\n  override var m24: Double\n  override
  var m31: Double\n  override var m32: Double\n  override var m33: Double\n  override var m34: Double\n
  override var m41: Double\n  override var m42: Double\n  override var m43: Double\n  override var m44:
  Double\n  fun multiplySelf(other: DOMMatrix): DOMMatrix\n  fun preMultiplySelf(other: DOMMatrix):
  DOMMatrix\n  fun translateSelf(tx: Double, ty: Double, tz: Double = definedExternally): DOMMatrix\n  fun
  scaleSelf(scale: Double, originX: Double = definedExternally, originY: Double = definedExternally): DOMMatrix\n
  fun scale3dSelf(scale: Double, originX: Double = definedExternally, originY: Double = definedExternally,
  originZ: Double = definedExternally): DOMMatrix\n  fun scaleNonUniformSelf(scaleX: Double, scaleY: Double =
  definedExternally, scaleZ: Double = definedExternally, originX: Double = definedExternally, originY: Double =
  definedExternally, originZ: Double = definedExternally): DOMMatrix\n  fun rotateSelf(angle: Double, originX:
  Double = definedExternally, originY: Double = definedExternally): DOMMatrix\n  fun rotateFromVectorSelf(x:
  Double, y: Double): DOMMatrix\n  fun rotateAxisAngleSelf(x: Double, y: Double, z: Double, angle: Double):
  DOMMatrix\n  fun skewXSelf(sx: Double): DOMMatrix\n  fun skewYSelf(sy: Double): DOMMatrix\n  fun
  invertSelf(): DOMMatrix\n  fun setMatrixValue(transformList: String): DOMMatrix\n}\n\npublic external
interface ScrollOptions {\n  var behavior: ScrollBehavior? /* = ScrollBehavior.AUTO */\n  get() =
  definedExternally\n  set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
  \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ScrollOptions(behavior:
  ScrollBehavior? = ScrollBehavior.AUTO): ScrollOptions {\n  val o = js(\"({})\")\n  o[\"behavior\"] = behavior\n
  return o\n}\n\n/**\n * Exposes the JavaScript
[ScrollToOptions](https://developer.mozilla.org/en/docs/Web/API/ScrollToOptions) to Kotlin\n
*/\npublic external

```



```

interface ScrollToOptions : ScrollOptions {
    var left: Double?
    get() = definedExternally
    set(value) = definedExternally
    var top: Double?
    get() = definedExternally
    set(value) = definedExternally
}
@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun ScrollToOptions(left: Double? = undefined, top: Double? = undefined, behavior: ScrollBehavior? = ScrollBehavior.AUTO): ScrollToOptions {
    val o = js("{}")
    o["left"] = left
    o["top"] = top
    o["behavior"] = behavior
    return o
}
/**
 * Exposes the JavaScript [MediaQueryList](https://developer.mozilla.org/en/docs/Web/API/MediaQueryList) to Kotlin
 */
public external abstract class MediaQueryList : EventTarget {
    open val media: String
    open val matches: Boolean
    open var onchange: ((Event) -> dynamic)?
    fun addListener(listener: EventListener?)
    fun addListener(listener: ((Event) -> Unit)?)
    fun removeListener(listener: EventListener?)
    fun removeListener(listener: ((Event) -> Unit)?)
}
/**
 * Exposes the JavaScript [MediaQueryListEvent](https://developer.mozilla.org/en/docs/Web/API/MediaQueryListEvent) to Kotlin
 */
public external open class MediaQueryListEvent(type: String, eventInitDict: MediaQueryListEventInit = definedExternally) : Event {
    open val media: String
    open val matches: Boolean
    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}
public external interface MediaQueryListEventInit : EventInit {
    var media: String? /* = "" */
    get() = definedExternally
    set(value) = definedExternally
    var matches: Boolean? /* = false */
    get() = definedExternally
    set(value) = definedExternally
}
@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun MediaQueryListEventInit(media: String? = "", matches: Boolean? = false, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): MediaQueryListEventInit {
    val o = js("{}")
    o["media"] = media
    o["matches"] = matches
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}
/**
 * Exposes the JavaScript [Screen](https://developer.mozilla.org/en/docs/Web/API/Screen) to Kotlin
 */
public external abstract class Screen {
    open val availWidth: Int
    open val availHeight: Int
    open val width: Int
    open val height: Int
    open val colorDepth: Int
    open val pixelDepth: Int
}
/**
 * Exposes the JavaScript [CaretPosition](https://developer.mozilla.org/en/docs/Web/API/CaretPosition) to Kotlin
 */
public external abstract class CaretPosition {
    open val offsetNode: Node
    open val offset: Int
    fun getClientRect(): DOMRect?
}
public external interface ScrollIntoViewOptions : ScrollOptions {
    var block: ScrollLogicalPosition? /* = ScrollLogicalPosition.CENTER */
    get() = definedExternally
    set(value) = definedExternally
    var inline: ScrollLogicalPosition? /* = ScrollLogicalPosition.CENTER */
    get() = definedExternally
    set(value) = definedExternally
}
@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun ScrollIntoViewOptions(block: ScrollLogicalPosition? = ScrollLogicalPosition.CENTER, inline: ScrollLogicalPosition? = ScrollLogicalPosition.CENTER, behavior: ScrollBehavior? = ScrollBehavior.AUTO): ScrollIntoViewOptions {
    val o = js("{}")
    o["block"] = block
    o["inline"] = inline
    o["behavior"] = behavior
    return o
}
public external interface BoxQuadOptions {
    var box: CSSBoxType? /* = CSSBoxType.BORDER */
    get() = definedExternally
    set(value) = definedExternally
    var relativeTo: dynamic
    get() = definedExternally
    set(value) = definedExternally
}
@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun BoxQuadOptions(box: CSSBoxType? = CSSBoxType.BORDER, relativeTo: dynamic = undefined): BoxQuadOptions {
    val o = js("{}")
    o["box"] = box
    o["relativeTo"] = relativeTo
    return o
}
public external interface ConvertCoordinateOptions {
    var fromBox: CSSBoxType? /* = CSSBoxType.BORDER */
    get() = definedExternally
    set(value) = definedExternally
    var toBox: CSSBoxType? /* = CSSBoxType.BORDER */
    get() = definedExternally
    set(value) = definedExternally
}
@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun ConvertCoordinateOptions(fromBox: CSSBoxType? = CSSBoxType.BORDER, toBox: CSSBoxType? = CSSBoxType.BORDER):

```

```

ConvertCoordinateOptions {\n  val o = js("{}")\n  o["fromBox"] = fromBox\n  o["toBox"] = toBox\n  return o}\n\n/**\n * Exposes the JavaScript
[GeometryUtils](https://developer.mozilla.org/en/docs/Web/API/GeometryUtils) to Kotlin\n *\npublic external
interface GeometryUtils {\n  fun getBoxQuads(options: BoxQuadOptions = definedExternally):
Array<DOMQuad>\n  fun convertQuadFromNode(quad: dynamic, from: dynamic, options:
ConvertCoordinateOptions = definedExternally): DOMQuad\n  fun convertRectFromNode(rect:
DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions = definedExternally): DOMQuad\n  fun
convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions =
definedExternally): DOMPoint}\n\n/**\n * Exposes the JavaScript
[Touch](https://developer.mozilla.org/en/docs/Web/API/Touch) to Kotlin\n *\npublic external abstract class Touch
{\n  open val identifier: Int\n  open val target: EventTarget\n  open val screenX: Int\n  open val screenY: Int\n
open val clientX: Int\n  open val clientY: Int\n  open val pageX: Int\n  open val pageY: Int\n  open val region:
String?\n}\n\npublic external abstract class TouchList : ItemArrayLike<Touch> {\n  override fun item(index: Int):
Touch?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun TouchList.get(index: Int):
Touch? = asDynamic()[index]\n\npublic external open class TouchEvent : UIEvent {\n  open val touches:
TouchList\n  open val targetTouches: TouchList\n  open val changedTouches: TouchList\n  open val altKey:
Boolean\n  open val metaKey: Boolean\n  open val ctrlKey: Boolean\n  open val shiftKey: Boolean\n\n  companion object {\n    val NONE: Short\n    val CAPTURING_PHASE: Short\n    val AT_TARGET:
Short\n    val BUBBLING_PHASE: Short\n  }\n}\n\n/**\n * Exposes the JavaScript
[Image](https://developer.mozilla.org/en/docs/Web/API/Image) to Kotlin\n *\npublic external open class
Image(width: Int = definedExternally, height: Int = definedExternally) : HTMLImageElement {\n  override var
onabort: ((Event) -> dynamic)?\n  override var onblur: ((FocusEvent) -> dynamic)?\n  override var oncancel:
((Event) -> dynamic)?\n  override var oncanplay: ((Event) -> dynamic)?\n  override var oncanplaythrough:
((Event) -> dynamic)?\n  override var onchange: ((Event) -> dynamic)?\n  override var onclick: ((MouseEvent) ->
dynamic)?\n  override var onclose: ((Event) -> dynamic)?\n  override var oncontextmenu: ((MouseEvent) ->
dynamic)?\n  override var oncuechange: ((Event) -> dynamic)?\n  override var ondblclick: ((MouseEvent) ->
dynamic)?\n  override var ondrag: ((DragEvent) -> dynamic)?\n  override var ondragend: ((DragEvent) ->
dynamic)?\n  override var ondragenter: ((DragEvent) -> dynamic)?\n  override var ondragexit: ((DragEvent) ->
dynamic)?\n  override var ondragleave: ((DragEvent) -> dynamic)?\n  override var ondragover: ((DragEvent) ->
dynamic)?\n  override var ondragstart: ((DragEvent) -> dynamic)?\n  override var ondrop: ((DragEvent) ->
dynamic)?\n  override var ondurationchange: ((Event) -> dynamic)?\n  override var onemptied: ((Event) ->
dynamic)?\n  override var onended: ((Event) -> dynamic)?\n  override var onerror: ((dynamic, String, Int, Int,
Any?) -> dynamic)?\n  override var onfocus: ((FocusEvent) -> dynamic)?\n  override var oninput: ((InputEvent) ->
dynamic)?\n  override var oninvalid: ((Event) -> dynamic)?\n  override var onkeydown: ((KeyboardEvent) ->
dynamic)?\n  override var onkeypress: ((KeyboardEvent) -> dynamic)?\n  override var onkeyup:
((KeyboardEvent) -> dynamic)?\n  override var onload: ((Event) -> dynamic)?\n  override var onloadeddata:
((Event) -> dynamic)?\n  override var onloadedmetadata: ((Event) -> dynamic)?\n  override var onloadend:
((Event) -> dynamic)?\n  override var onloadstart: ((ProgressEvent) -> dynamic)?\n  override var onmousedown:
((MouseEvent) -> dynamic)?\n  override var onmouseenter: ((MouseEvent) -> dynamic)?\n  override var
onmouseleave: ((MouseEvent) -> dynamic)?\n  override var onmousemove: ((MouseEvent) -> dynamic)?\n
override var onmouseout: ((MouseEvent) -> dynamic)?\n  override var onmouseover: ((MouseEvent) ->
dynamic)?\n  override var onmouseup: ((MouseEvent) -> dynamic)?\n  override var onwheel: ((WheelEvent) ->
dynamic)?\n  override var onpause: ((Event) -> dynamic)?\n  override var onplay: ((Event) -> dynamic)?\n
override var onplaying: ((Event) -> dynamic)?\n  override var onprogress: ((ProgressEvent) -> dynamic)?\n
override var onratechange: ((Event) -> dynamic)?\n  override var onreset: ((Event) -> dynamic)?\n  override var
onresize: ((Event) -> dynamic)?\n  override var onscroll: ((Event) -> dynamic)?\n  override var onseeked:
((Event) -> dynamic)?\n  override var onseeking: ((Event) -> dynamic)?\n  override var onselect: ((Event) ->

```

```

dynamic)?\n  override var onshow: ((Event) -> dynamic)?\n  override var onstalled: ((Event) -> dynamic)?\n
override var onsubmit: ((Event) -> dynamic)?\n  override var onsuspend: ((Event) -> dynamic)?\n  override var
ontimeupdate: ((Event) -> dynamic)?\n  override var ontoggle: ((Event) -> dynamic)?\n  override var
onvolumechange: ((Event) -> dynamic)?\n  override var onwaiting: ((Event) -> dynamic)?\n  override var
ongotpointercapture: ((PointerEvent) -> dynamic)?\n  override var onlostpointercapture: ((PointerEvent) ->
dynamic)?\n  override var onpointerdown: ((PointerEvent) -> dynamic)?\n  override var onpointermove:
((PointerEvent) -> dynamic)?\n  override var onpointerup: ((PointerEvent) -> dynamic)?\n  override var
onpointercancel: ((PointerEvent) -> dynamic)?\n  override var onpointerover: ((PointerEvent) -> dynamic)?\n
override var onpointerout: ((PointerEvent) -> dynamic)?\n  override var onpointerenter: ((PointerEvent) ->
dynamic)?\n  override var onpointerleave: ((PointerEvent) -> dynamic)?\n  override var oncopy:
((ClipboardEvent) -> dynamic)?\n  override var oncut: ((ClipboardEvent) -> dynamic)?\n  override var onpaste:
((ClipboardEvent) -> dynamic)?\n  override var contentEditable: String\n  override val isContentEditable:
Boolean\n  override val style: CSSStyleDeclaration\n  override val children: HTMLCollection\n  override val
firstElementChild: Element?\n  override val lastElementChild: Element?\n  override val childElementCount: Int\n
  override val previousElementSibling: Element?\n  override val nextElementSibling: Element?\n  override val
assignedSlot: HTMLSlotElement?\n  override fun prepend(vararg nodes: dynamic)\n  override fun append(vararg
nodes: dynamic)\n  override fun querySelector(selectors: String): Element?\n  override fun
querySelectorAll(selectors: String): NodeList\n  override fun before(vararg nodes: dynamic)\n  override fun
after(vararg nodes: dynamic)\n  override fun replaceWith(vararg nodes: dynamic)\n  override fun remove()\n
override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad>\n  override
fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions /* =
definedExternally */): DOMQuad\n  override fun convertRectFromNode(rect: DOMRectReadOnly, from:
dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n  override fun
convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* =
definedExternally */): DOMPoint\n\n  companion object {\n    val ELEMENT_NODE: Short\n    val
ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val
ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n\n  public external open class
Audio(src: String = definedExternally) : HTMLAudioElement {\n    override var onabort: ((Event) -> dynamic)?\n
override var onblur: ((FocusEvent) -> dynamic)?\n  override var oncancel: ((Event) -> dynamic)?\n  override var
oncanplay: ((Event) -> dynamic)?\n  override var oncanplaythrough: ((Event) -> dynamic)?\n  override var
onchange: ((Event) -> dynamic)?\n  override var onclick: ((MouseEvent) -> dynamic)?\n  override var onclose:
((Event) -> dynamic)?\n  override var oncontextmenu: ((MouseEvent) -> dynamic)?\n  override var oncuechange:
((Event) -> dynamic)?\n  override var ondblclick: ((MouseEvent) -> dynamic)?\n  override var ondrag:
((DragEvent) -> dynamic)?\n  override var ondragend: ((DragEvent) -> dynamic)?\n  override var ondragenter:
((DragEvent) -> dynamic)?\n  override var ondragexit: ((DragEvent) -> dynamic)?\n  override var ondragleave:
((DragEvent) -> dynamic)?\n  override var ondragover: ((DragEvent) -> dynamic)?\n  override var ondragstart:
((DragEvent) -> dynamic)?\n  override var ondrop: ((DragEvent) -> dynamic)?\n  override var ondurationchange:
((Event) -> dynamic)?\n  override var onemptied: ((Event) -> dynamic)?\n  override var onended: ((Event) ->
dynamic)?\n  override var onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)?\n  override var onfocus:
((FocusEvent) -> dynamic)?\n  override var oninput: ((InputEvent) -> dynamic)?\n  override var oninvalid:
((Event) -> dynamic)?\n  override var onkeydown: ((KeyboardEvent) -> dynamic)?\n  override var onkeypress:

```

```

((KeyboardEvent) -> dynamic)?\n  override var onkeyup: ((KeyboardEvent) -> dynamic)?\n  override var onload:
((Event) -> dynamic)?\n  override var onloadeddata: ((Event) -> dynamic)?\n  override var onloadedmetadata:
((Event) -> dynamic)?\n  override var onloadend: ((Event) -> dynamic)?\n  override var onloadstart:
((ProgressEvent) -> dynamic)?\n  override var onmousedown: ((MouseEvent) -> dynamic)?\n  override var
onmouseenter: ((MouseEvent) -> dynamic)?\n  override var onmouseleave: ((MouseEvent) -> dynamic)?\n
override var onmousemove: ((MouseEvent) -> dynamic)?\n  override var onmouseout: ((MouseEvent) ->
dynamic)?\n  override var onmouseover: ((MouseEvent) -> dynamic)?\n  override var onmouseup: ((MouseEvent)
-> dynamic)?\n  override var onwheel: ((WheelEvent) -> dynamic)?\n  override var onpause: ((Event) ->
dynamic)?\n  override var onplay: ((Event) -> dynamic)?\n  override var onplaying: ((Event) -> dynamic)?\n
override var onprogress: ((ProgressEvent) -> dynamic)?\n  override var onratechange: ((Event) -> dynamic)?\n
override var onreset: ((Event) -> dynamic)?\n  override var onresize: ((Event) -> dynamic)?\n  override var
onscroll: ((Event) -> dynamic)?\n  override var onseeked: ((Event) -> dynamic)?\n  override var onseeking:
((Event) -> dynamic)?\n  override var onselect: ((Event) -> dynamic)?\n  override var onshow: ((Event) ->
dynamic)?\n  override var onstalled: ((Event) -> dynamic)?\n  override var onsubmit: ((Event) -> dynamic)?\n
override var onsuspend: ((Event) -> dynamic)?\n  override var ontimeupdate: ((Event) -> dynamic)?\n  override
var ontoggle: ((Event) -> dynamic)?\n  override var onvolumechange: ((Event) -> dynamic)?\n  override var
onwaiting: ((Event) -> dynamic)?\n  override var ongotpointercapture: ((PointerEvent) -> dynamic)?\n  override
var onlostpointercapture: ((PointerEvent) -> dynamic)?\n  override var onpointerdown: ((PointerEvent) ->
dynamic)?\n  override var onpointermove: ((PointerEvent) -> dynamic)?\n  override var onpointerup:
((PointerEvent) -> dynamic)?\n  override var onpointercancel: ((PointerEvent) -> dynamic)?\n  override var
onpointerover: ((PointerEvent) -> dynamic)?\n  override var onpointerout: ((PointerEvent) -> dynamic)?\n
override var onpointerenter: ((PointerEvent) -> dynamic)?\n  override var onpointerleave: ((PointerEvent) ->
dynamic)?\n  override var oncopy: ((ClipboardEvent) -> dynamic)?\n  override var oncut: ((ClipboardEvent) ->
dynamic)?\n  override var onpaste: ((ClipboardEvent) -> dynamic)?\n  override var contentEditable: String\n
override val isContentEditable: Boolean\n  override val style: CSSStyleDeclaration\n  override val children:
HTMLCollection\n  override val firstElementChild: Element?\n  override val lastElementChild: Element?\n
override val childElementCount: Int\n  override val previousElementSibling: Element?\n  override val
nextElementSibling: Element?\n  override val assignedSlot: HTMLSlotElement?\n  override fun prepend(vararg
nodes: dynamic)\n  override fun append(vararg nodes: dynamic)\n  override fun querySelector(selectors: String):
Element?\n  override fun querySelectorAll(selectors: String): NodeList\n  override fun before(vararg nodes:
dynamic)\n  override fun after(vararg nodes: dynamic)\n  override fun replaceWith(vararg nodes: dynamic)\n
override fun remove()\n  override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */):
Array<DOMQuad>\n  override fun convertQuadFromNode(quad: dynamic, from: dynamic, options:
ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n  override fun convertRectFromNode(rect:
DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n
override fun convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* =
definedExternally */): DOMPoint\n\n  companion object {\n    val NETWORK_EMPTY: Short\n    val
NETWORK_IDLE: Short\n    val NETWORK_LOADING: Short\n    val NETWORK_NO_SOURCE: Short\n
    val HAVE_NOTHING: Short\n    val HAVE_METADATA: Short\n    val HAVE_CURRENT_DATA:
Short\n    val HAVE_FUTURE_DATA: Short\n    val HAVE_ENOUGH_DATA: Short\n    val
ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val
CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE:
Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val

```



```

after(vararg nodes: dynamic)\n  override fun replaceWith(vararg nodes: dynamic)\n  override fun remove()\n  override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad>\n  override fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n  override fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n  override fun convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint\n\n  companion object {\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n    val DOCUMENT_POSITION_CONTAINED_BY: Short\n  }\n\n  public external interface UnionElementOrHTMLCollection\n  public external interface UnionElementOrRadioNodeList\n  public external interface UnionHTMLOptGroupElementOrHTMLOptionElement\n  public external interface UnionAudioTrackOrTextTrackOrVideoTrack\n  public external interface UnionElementOrMouseEvent\n  public external interface UnionMessagePortOrWindowProxy\n  public external interface MediaProvider\n  public external interface RenderingContext\n  public external interface HTMLOrSVGImageElement : CanvasImageSource\n  public external interface CanvasImageSource : ImageBitmapSource\n  public external interface ImageBitmapSource\n  public external interface HTMLOrSVGScriptElement\n\n  /* please, don't implement this interface!\n  *\n  @JsName("null")\n  @Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n  public external interface DocumentReadyState {\n    companion object\n  }\n  public inline val DocumentReadyState.Companion.LOADING: DocumentReadyState get() = "loading".asDynamic().unsafeCast<DocumentReadyState>()\n  public inline val DocumentReadyState.Companion.INTERACTIVE: DocumentReadyState get() = "interactive".asDynamic().unsafeCast<DocumentReadyState>()\n  public inline val DocumentReadyState.Companion.COMPLETE: DocumentReadyState get() = "complete".asDynamic().unsafeCast<DocumentReadyState>()\n\n  /* please, don't implement this interface!\n  *\n  @JsName("null")\n  @Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n  public external interface CanPlayTypeResult {\n    companion object\n  }\n  public inline val CanPlayTypeResult.Companion.EMPTY: CanPlayTypeResult get() = "".asDynamic().unsafeCast<CanPlayTypeResult>()\n  public inline val CanPlayTypeResult.Companion.MAYBE: CanPlayTypeResult get() = "maybe".asDynamic().unsafeCast<CanPlayTypeResult>()\n  public inline val CanPlayTypeResult.Companion.PROBABLY: CanPlayTypeResult get() = "probably".asDynamic().unsafeCast<CanPlayTypeResult>()\n\n  /* please, don't implement this interface!\n  *\n  @JsName("null")\n  @Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n  public external interface TextTrackMode {\n    companion object\n  }\n  public inline val TextTrackMode.Companion.DISABLED: TextTrackMode get() = "disabled".asDynamic().unsafeCast<TextTrackMode>()\n  public inline val TextTrackMode.Companion.HIDDEN: TextTrackMode get() = "hidden".asDynamic().unsafeCast<TextTrackMode>()\n  public inline val TextTrackMode.Companion.SHOWING: TextTrackMode get() = "showing".asDynamic().unsafeCast<TextTrackMode>()\n\n  /* please, don't implement this interface!\n  *\n  @JsName("null")\n  @Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n  public external interface TextTrackKind {\n    companion object\n  }\n  public inline val TextTrackKind.Companion.SUBTITLES:

```

```

TextTrackKind get() = \"subtitles\".asDynamic().unsafeCast<TextTrackKind>()\n\npublic inline val
TextTrackKind.Companion.CAPTIONS: TextTrackKind get() =
\"captions\".asDynamic().unsafeCast<TextTrackKind>()\n\npublic inline val
TextTrackKind.Companion.DESCRPTIONS: TextTrackKind get() =
\"descriptions\".asDynamic().unsafeCast<TextTrackKind>()\n\npublic inline val
TextTrackKind.Companion.CHAPTERS: TextTrackKind get() =
\"chapters\".asDynamic().unsafeCast<TextTrackKind>()\n\npublic inline val
TextTrackKind.Companion.METADATA: TextTrackKind get() =
\"metadata\".asDynamic().unsafeCast<TextTrackKind>()\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface SelectionMode {\n\n    companion object\n\n}\n\npublic inline val SelectionMode.Companion.SELECT:
SelectionMode get() = \"select\".asDynamic().unsafeCast<SelectionMode>()\n\npublic inline val
SelectionMode.Companion.START: SelectionMode get() =
\"start\".asDynamic().unsafeCast<SelectionMode>()\n\npublic inline val SelectionMode.Companion.END:
SelectionMode get() = \"end\".asDynamic().unsafeCast<SelectionMode>()\n\npublic inline val
SelectionMode.Companion.PRESERVE: SelectionMode get() =
\"preserve\".asDynamic().unsafeCast<SelectionMode>()\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface CanvasFillRule {\n\n    companion object\n\n}\n\npublic inline val CanvasFillRule.Companion.NONZERO:
CanvasFillRule get() = \"nonzero\".asDynamic().unsafeCast<CanvasFillRule>()\n\npublic inline val
CanvasFillRule.Companion.EVENODD: CanvasFillRule get() =
\"evenodd\".asDynamic().unsafeCast<CanvasFillRule>()\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface ImageSmoothingQuality {\n\n    companion object\n\n}\n\npublic inline val
ImageSmoothingQuality.Companion.LOW: ImageSmoothingQuality get() =
\"low\".asDynamic().unsafeCast<ImageSmoothingQuality>()\n\npublic inline val
ImageSmoothingQuality.Companion.MEDIUM: ImageSmoothingQuality get() =
\"medium\".asDynamic().unsafeCast<ImageSmoothingQuality>()\n\npublic inline val
ImageSmoothingQuality.Companion.HIGH: ImageSmoothingQuality get() =
\"high\".asDynamic().unsafeCast<ImageSmoothingQuality>()\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface CanvasLineCap {\n\n    companion object\n\n}\n\npublic inline val CanvasLineCap.Companion.BUTT:
CanvasLineCap get() = \"butt\".asDynamic().unsafeCast<CanvasLineCap>()\n\npublic inline val
CanvasLineCap.Companion.ROUND: CanvasLineCap get() =
\"round\".asDynamic().unsafeCast<CanvasLineCap>()\n\npublic inline val CanvasLineCap.Companion.SQUARE:
CanvasLineCap get() = \"square\".asDynamic().unsafeCast<CanvasLineCap>()\n\n/* please, don't implement this
interface! *\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic
external interface CanvasLineJoin {\n\n    companion object\n\n}\n\npublic inline val
CanvasLineJoin.Companion.ROUND: CanvasLineJoin get() =
\"round\".asDynamic().unsafeCast<CanvasLineJoin>()\n\npublic inline val CanvasLineJoin.Companion.BEVEL:
CanvasLineJoin get() = \"bevel\".asDynamic().unsafeCast<CanvasLineJoin>()\n\npublic inline val
CanvasLineJoin.Companion.MITER: CanvasLineJoin get() =
\"miter\".asDynamic().unsafeCast<CanvasLineJoin>()\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface CanvasTextAlign {\n\n    companion object\n\n}\n\npublic inline val CanvasTextAlign.Companion.START:
CanvasTextAlign get() = \"start\".asDynamic().unsafeCast<CanvasTextAlign>()\n\npublic inline val
CanvasTextAlign.Companion.END: CanvasTextAlign get() =
\"end\".asDynamic().unsafeCast<CanvasTextAlign>()\n\npublic inline val CanvasTextAlign.Companion.LEFT:

```

```

CanvasTextAlign get() = "left".asDynamic().unsafeCast<CanvasTextAlign>()\n\npublic inline val
CanvasTextAlign.Companion.RIGHT: CanvasTextAlign get() =
"right".asDynamic().unsafeCast<CanvasTextAlign>()\n\npublic inline val
CanvasTextAlign.Companion.CENTER: CanvasTextAlign get() =
"center".asDynamic().unsafeCast<CanvasTextAlign>()\n\n/* please, don't implement this interface!
*\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external
interface CanvasTextBaseline {\n  companion object\n}\n\npublic inline val CanvasTextBaseline.Companion.TOP:
CanvasTextBaseline get() = "top".asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.HANGING: CanvasTextBaseline get() =
"hanging".asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.MIDDLE: CanvasTextBaseline get() =
"middle".asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.ALPHABETIC: CanvasTextBaseline get() =
"alphabetic".asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.IDEOGRAPHIC: CanvasTextBaseline get() =
"ideographic".asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.BOTTOM: CanvasTextBaseline get() =
"bottom".asDynamic().unsafeCast<CanvasTextBaseline>()\n\n/* please, don't implement this interface!
*\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external
interface CanvasDirection {\n  companion object\n}\n\npublic inline val CanvasDirection.Companion.LTR:
CanvasDirection get() = "ltr".asDynamic().unsafeCast<CanvasDirection>()\n\npublic inline val
CanvasDirection.Companion.RTL: CanvasDirection get() =
"rtl".asDynamic().unsafeCast<CanvasDirection>()\n\npublic inline val CanvasDirection.Companion.INHERIT:
CanvasDirection get() = "inherit".asDynamic().unsafeCast<CanvasDirection>()\n\n/* please, don't implement this
interface! *\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic
external interface ScrollRestoration {\n  companion object\n}\n\npublic inline val
ScrollRestoration.Companion.AUTO: ScrollRestoration get() =
"auto".asDynamic().unsafeCast<ScrollRestoration>()\n\npublic inline val
ScrollRestoration.Companion.MANUAL: ScrollRestoration get() =
"manual".asDynamic().unsafeCast<ScrollRestoration>()\n\n/* please, don't implement this interface!
*\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external
interface ImageOrientation {\n  companion object\n}\n\npublic inline val ImageOrientation.Companion.NONE:
ImageOrientation get() = "none".asDynamic().unsafeCast<ImageOrientation>()\n\npublic inline val
ImageOrientation.Companion.FLIPY: ImageOrientation get() =
"flipY".asDynamic().unsafeCast<ImageOrientation>()\n\n/* please, don't implement this interface!
*\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external
interface PremultiplyAlpha {\n  companion object\n}\n\npublic inline val PremultiplyAlpha.Companion.NONE:
PremultiplyAlpha get() = "none".asDynamic().unsafeCast<PremultiplyAlpha>()\n\npublic inline val
PremultiplyAlpha.Companion.PREMULTIPLY: PremultiplyAlpha get() =
"premultiply".asDynamic().unsafeCast<PremultiplyAlpha>()\n\npublic inline val
PremultiplyAlpha.Companion.DEFAULT: PremultiplyAlpha get() =
"default".asDynamic().unsafeCast<PremultiplyAlpha>()\n\n/* please, don't implement this interface!
*\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external
interface ColorSpaceConversion {\n  companion object\n}\n\npublic inline val
ColorSpaceConversion.Companion.NONE: ColorSpaceConversion get() =
"none".asDynamic().unsafeCast<ColorSpaceConversion>()\n\npublic inline val
ColorSpaceConversion.Companion.DEFAULT: ColorSpaceConversion get() =
"default".asDynamic().unsafeCast<ColorSpaceConversion>()\n\n/* please, don't implement this interface!

```



```

*\n@jsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface ResizeQuality {\n  companion object\n}\n\npublic inline val ResizeQuality.Companion.PIXELATED:
ResizeQuality get() = "pixelated".asDynamic().unsafeCast<ResizeQuality>()\n\npublic inline val
ResizeQuality.Companion.LOW: ResizeQuality get() =
"low".asDynamic().unsafeCast<ResizeQuality>()\n\npublic inline val ResizeQuality.Companion.MEDIUM:
ResizeQuality get() = "medium".asDynamic().unsafeCast<ResizeQuality>()\n\npublic inline val
ResizeQuality.Companion.HIGH: ResizeQuality get() = "high".asDynamic().unsafeCast<ResizeQuality>()\n\n/*
please, don't implement this interface!
*\n@jsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface BinaryType {\n  companion object\n}\n\npublic inline val BinaryType.Companion.BLOB: BinaryType
get() = "blob".asDynamic().unsafeCast<BinaryType>()\n\npublic inline val
BinaryType.Companion.ARRAYBUFFER: BinaryType get() =
"arraybuffer".asDynamic().unsafeCast<BinaryType>()\n\n/* please, don't implement this interface!
*\n@jsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface WorkerType {\n  companion object\n}\n\npublic inline val WorkerType.Companion.CLASSIC:
WorkerType get() = "classic".asDynamic().unsafeCast<WorkerType>()\n\npublic inline val
WorkerType.Companion.MODULE: WorkerType get() =
"module".asDynamic().unsafeCast<WorkerType>()\n\n/* please, don't implement this interface!
*\n@jsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface ShadowRootMode {\n  companion object\n}\n\npublic inline val ShadowRootMode.Companion.OPEN:
ShadowRootMode get() = "open".asDynamic().unsafeCast<ShadowRootMode>()\n\npublic inline val
ShadowRootMode.Companion.CLOSED: ShadowRootMode get() =
"closed".asDynamic().unsafeCast<ShadowRootMode>()\n\n/* please, don't implement this interface!
*\n@jsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface ScrollBehavior {\n  companion object\n}\n\npublic inline val ScrollBehavior.Companion.AUTO:
ScrollBehavior get() = "auto".asDynamic().unsafeCast<ScrollBehavior>()\n\npublic inline val
ScrollBehavior.Companion.INSTANT: ScrollBehavior get() =
"instant".asDynamic().unsafeCast<ScrollBehavior>()\n\npublic inline val ScrollBehavior.Companion.SMOOTH:
ScrollBehavior get() = "smooth".asDynamic().unsafeCast<ScrollBehavior>()\n\n/* please, don't implement this
interface! *\n@jsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic
external interface ScrollLogicalPosition {\n  companion object\n}\n\npublic inline val
ScrollLogicalPosition.Companion.START: ScrollLogicalPosition get() =
"start".asDynamic().unsafeCast<ScrollLogicalPosition>()\n\npublic inline val
ScrollLogicalPosition.Companion.CENTER: ScrollLogicalPosition get() =
"center".asDynamic().unsafeCast<ScrollLogicalPosition>()\n\npublic inline val
ScrollLogicalPosition.Companion.END: ScrollLogicalPosition get() =
"end".asDynamic().unsafeCast<ScrollLogicalPosition>()\n\npublic inline val
ScrollLogicalPosition.Companion.NEAREST: ScrollLogicalPosition get() =
"nearest".asDynamic().unsafeCast<ScrollLogicalPosition>()\n\n/* please, don't implement this interface!
*\n@jsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface CSSBoxType {\n  companion object\n}\n\npublic inline val CSSBoxType.Companion.MARGIN:
CSSBoxType get() = "margin".asDynamic().unsafeCast<CSSBoxType>()\n\npublic inline val
CSSBoxType.Companion.BORDER: CSSBoxType get() =
"border".asDynamic().unsafeCast<CSSBoxType>()\n\npublic inline val CSSBoxType.Companion.PADDING:
CSSBoxType get() = "padding".asDynamic().unsafeCast<CSSBoxType>()\n\npublic inline val
CSSBoxType.Companion.CONTENT: CSSBoxType get() =
"content".asDynamic().unsafeCast<CSSBoxType>()"/\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be

```

```

found in the license/LICENSE.txt file.\n *\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n\n//
See github.com/kotlin/dukat for details\n\npackage org.w3c.fetch\n\nimport kotlin.js.*\nimport
org.khronos.webgl.*\nimport org.w3c.files.*\nimport org.w3c.xhr.*\n\n/**\n * Exposes the JavaScript
[Headers](https://developer.mozilla.org/en/docs/Web/API/Headers) to Kotlin\n *\npublic external open class
Headers(init: dynamic = definedExternally) {\n    fun append(name: String, value: String)\n    fun delete(name:
String)\n    fun get(name: String): String?\n    fun has(name: String): Boolean\n    fun set(name: String, value:
String)\n}\n\n/**\n * Exposes the JavaScript [Body](https://developer.mozilla.org/en/docs/Web/API/Body) to
Kotlin\n *\npublic external interface Body {\n    val bodyUsed: Boolean\n    fun arrayBuffer():
Promise<ArrayBuffer>\n    fun blob(): Promise<Blob>\n    fun formData(): Promise<FormData>\n    fun json():
Promise<Any?>\n    fun text(): Promise<String>\n}\n\n/**\n * Exposes the JavaScript
[Request](https://developer.mozilla.org/en/docs/Web/API/Request) to Kotlin\n *\npublic external open class
Request(input: dynamic, init: RequestInit = definedExternally) : Body {\n    open val method: String\n    open val
url: String\n    open val headers: Headers\n    open val type: RequestType\n    open val destination:
RequestDestination\n    open val referrer: String\n    open val referrerPolicy: dynamic\n    open val mode:
RequestMode\n    open val credentials: RequestCredentials\n    open val cache: RequestCache\n    open val redirect:
RequestRedirect\n    open val integrity: String\n    open val keepalive: Boolean\n    override val bodyUsed:
Boolean\n    fun clone(): Request\n    override fun arrayBuffer(): Promise<ArrayBuffer>\n    override fun blob():
Promise<Blob>\n    override fun formData(): Promise<FormData>\n    override fun json(): Promise<Any?>\n    override
fun text(): Promise<String>\n}\n\npublic external interface RequestInit {\n    var method: String?\n    get() =
definedExternally\n    set(value) = definedExternally\n    var headers: dynamic\n    get() =
definedExternally\n    set(value) = definedExternally\n    var body: dynamic\n    get() = definedExternally\n
set(value) = definedExternally\n    var referrer: String?\n    get() = definedExternally\n    set(value) =
definedExternally\n    var referrerPolicy: dynamic\n    get() = definedExternally\n    set(value) =
definedExternally\n    var mode: RequestMode?\n    get() = definedExternally\n    set(value) =
definedExternally\n    var credentials: RequestCredentials?\n    get() = definedExternally\n    set(value) =
definedExternally\n    var cache: RequestCache?\n    get() = definedExternally\n    set(value) =
definedExternally\n    var redirect: RequestRedirect?\n    get() = definedExternally\n    set(value) =
definedExternally\n    var integrity: String?\n    get() = definedExternally\n    set(value) = definedExternally\n
var keepalive: Boolean?\n    get() = definedExternally\n    set(value) = definedExternally\n    var window:
Any?\n    get() = definedExternally\n    set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun RequestInit(method: String? =
undefined, headers: dynamic = undefined, body: dynamic = undefined, referrer: String? = undefined, referrerPolicy:
dynamic = undefined, mode: RequestMode? = undefined, credentials: RequestCredentials? = undefined, cache:
RequestCache? = undefined, redirect: RequestRedirect? = undefined, integrity: String? = undefined, keepalive:
Boolean? = undefined, window: Any? = undefined): RequestInit {\n    val o = js(\"({})\")\n    o[\"method\"] =
method\n    o[\"headers\"] = headers\n    o[\"body\"] = body\n    o[\"referrer\"] = referrer\n    o[\"referrerPolicy\"] =
referrerPolicy\n    o[\"mode\"] = mode\n    o[\"credentials\"] = credentials\n    o[\"cache\"] = cache\n    o[\"redirect\"]
= redirect\n    o[\"integrity\"] = integrity\n    o[\"keepalive\"] = keepalive\n    o[\"window\"] = window\n    return
o\n}\n\n/**\n * Exposes the JavaScript [Response](https://developer.mozilla.org/en/docs/Web/API/Response) to
Kotlin\n *\npublic external open class Response(body: dynamic = definedExternally, init: ResponseInit =
definedExternally) : Body {\n    open val type: ResponseType\n    open val url: String\n    open val redirected:
Boolean\n    open val status: Short\n    open val ok: Boolean\n    open val statusText: String\n    open val headers:
Headers\n    open val body: dynamic\n    open val trailer: Promise<Headers>\n    override val bodyUsed: Boolean\n
fun clone(): Response\n    override fun arrayBuffer(): Promise<ArrayBuffer>\n    override fun blob():
Promise<Blob>\n    override fun formData(): Promise<FormData>\n    override fun json(): Promise<Any?>\n    override
fun text(): Promise<String>\n\n    companion object {\n        fun error(): Response\n        fun redirect(url:
String, status: Short = definedExternally): Response\n    }\n}\n\npublic external interface ResponseInit {\n    var

```

```

status: Short? /* = 200 */\n    get() = definedExternally\n    set(value) = definedExternally\n    var statusText:
String? /* = \"OK\" */\n    get() = definedExternally\n    set(value) = definedExternally\n    var headers:
dynamic\n    get() = definedExternally\n    set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ResponseInit(status: Short? = 200,
statusText: String? = \"OK\", headers: dynamic = undefined): ResponseInit {\n    val o = js(\"({})\")\n    o[\"status\"]
= status\n    o[\"statusText\"] = statusText\n    o[\"headers\"] = headers\n    return o\n}\n\n/* please, don't implement
this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic
external interface RequestType {\n    companion object\n}\n\npublic inline val RequestType.Companion.EMPTY:
RequestType get() = \"\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val
RequestType.Companion.AUDIO: RequestType get() =
\"audio\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val RequestType.Companion.FONT:
RequestType get() = \"font\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val
RequestType.Companion.IMAGE: RequestType get() =
\"image\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val RequestType.Companion.SCRIPT:
RequestType get() = \"script\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val
RequestType.Companion.STYLE: RequestType get() =
\"style\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val RequestType.Companion.TRACK:
RequestType get() = \"track\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val
RequestType.Companion.VIDEO: RequestType get() = \"video\".asDynamic().unsafeCast<RequestType>()\n\n/*
please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface RequestDestination {\n    companion object\n}\n\npublic inline val
RequestDestination.Companion.EMPTY: RequestDestination get() =
\"\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.DOCUMENT: RequestDestination get() =
\"document\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.EMBED: RequestDestination get() =
\"embed\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.FONT: RequestDestination get() =
\"font\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.IMAGE: RequestDestination get() =
\"image\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.MANIFEST: RequestDestination get() =
\"manifest\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.MEDIA: RequestDestination get() =
\"media\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.OBJECT: RequestDestination get() =
\"object\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.REPORT: RequestDestination get() =
\"report\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.SCRIPT: RequestDestination get() =
\"script\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.SERVICEMANIFEST: RequestDestination get() =
\"servicemanifest\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.SERVICEMANIFEST: RequestDestination get() =
\"servicemanifest\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.SHAREDWORKER: RequestDestination get() =
\"sharedworker\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.STYLE: RequestDestination get() =

```

```

\"style\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.WORKER: RequestDestination get() =
\"worker\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.XSLT: RequestDestination get() =
\"xslt\".asDynamic().unsafeCast<RequestDestination>()\n\n/* please, don't implement this interface!
*\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface RequestMode {\n  companion object\n}\n\npublic inline val RequestMode.Companion.NAVIGATE:
RequestMode get() = \"navigate\".asDynamic().unsafeCast<RequestMode>()\n\npublic inline val
RequestMode.Companion.SAME_ORIGIN: RequestMode get() = \"same-
origin\".asDynamic().unsafeCast<RequestMode>()\n\npublic inline val RequestMode.Companion.NO_CORS:
RequestMode get() = \"no-cors\".asDynamic().unsafeCast<RequestMode>()\n\npublic inline val
RequestMode.Companion.CORS: RequestMode get() = \"cors\".asDynamic().unsafeCast<RequestMode>()\n\n/*
please, don't implement this interface!
*\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface RequestCredentials {\n  companion object\n}\n\npublic inline val RequestCredentials.Companion.OMIT:
RequestCredentials get() = \"omit\".asDynamic().unsafeCast<RequestCredentials>()\n\npublic inline val
RequestCredentials.Companion.SAME_ORIGIN: RequestCredentials get() = \"same-
origin\".asDynamic().unsafeCast<RequestCredentials>()\n\npublic inline val
RequestCredentials.Companion.INCLUDE: RequestCredentials get() =
\"include\".asDynamic().unsafeCast<RequestCredentials>()\n\n/* please, don't implement this interface!
*\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface RequestCache {\n  companion object\n}\n\npublic inline val RequestCache.Companion.DEFAULT:
RequestCache get() = \"default\".asDynamic().unsafeCast<RequestCache>()\n\npublic inline val
RequestCache.Companion.NO_STORE: RequestCache get() = \"no-
store\".asDynamic().unsafeCast<RequestCache>()\n\npublic inline val RequestCache.Companion.RELOAD:
RequestCache get() = \"reload\".asDynamic().unsafeCast<RequestCache>()\n\npublic inline val
RequestCache.Companion.NO_CACHE: RequestCache get() = \"no-
cache\".asDynamic().unsafeCast<RequestCache>()\n\npublic inline val
RequestCache.Companion.FORCE_CACHE: RequestCache get() = \"force-
cache\".asDynamic().unsafeCast<RequestCache>()\n\npublic inline val
RequestCache.Companion.ONLY_IF_CACHED: RequestCache get() = \"only-if-
cached\".asDynamic().unsafeCast<RequestCache>()\n\n/* please, don't implement this interface!
*\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface RequestRedirect {\n  companion object\n}\n\npublic inline val RequestRedirect.Companion.FOLLOW:
RequestRedirect get() = \"follow\".asDynamic().unsafeCast<RequestRedirect>()\n\npublic inline val
RequestRedirect.Companion.ERROR: RequestRedirect get() =
\"error\".asDynamic().unsafeCast<RequestRedirect>()\n\npublic inline val RequestRedirect.Companion.MANUAL:
RequestRedirect get() = \"manual\".asDynamic().unsafeCast<RequestRedirect>()\n\n/* please, don't implement this
interface! *\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic
external interface ResponseType {\n  companion object\n}\n\npublic inline val ResponseType.Companion.BASIC:
ResponseType get() = \"basic\".asDynamic().unsafeCast<ResponseType>()\n\npublic inline val
ResponseType.Companion.CORS: ResponseType get() =
\"cors\".asDynamic().unsafeCast<ResponseType>()\n\npublic inline val ResponseType.Companion.DEFAULT:
ResponseType get() = \"default\".asDynamic().unsafeCast<ResponseType>()\n\npublic inline val
ResponseType.Companion.ERROR: ResponseType get() =
\"error\".asDynamic().unsafeCast<ResponseType>()\n\npublic inline val ResponseType.Companion.OPAQUE:
ResponseType get() = \"opaque\".asDynamic().unsafeCast<ResponseType>()\n\npublic inline val
ResponseType.Companion.OPAQUEREDIRECT: ResponseType get() =

```

```

\opaqueredirect\).asDynamic().unsafeCast<ResponseType>()", /*\n * Copyright 2010-2021 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT
EDIT!\n\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.dom.mediacapture\n\nimport
kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\n\n/**\n * Exposes
the JavaScript [MediaStream](https://developer.mozilla.org/en/docs/Web/API/MediaStream) to Kotlin\n */\npublic
external open class MediaStream() : EventTarget, MediaProvider {\n    constructor(stream: MediaStream)\n
constructor(tracks: Array<MediaStreamTrack>)\n    open val id: String\n    open val active: Boolean\n    var
onaddtrack: ((MediaStreamTrackEvent) -> dynamic)?\n    var onremovetrack: ((MediaStreamTrackEvent) ->
dynamic)?\n    fun getAudioTracks(): Array<MediaStreamTrack>\n    fun getVideoTracks():
Array<MediaStreamTrack>\n    fun getTracks(): Array<MediaStreamTrack>\n    fun getTrackById(trackId: String):
MediaStreamTrack?\n    fun addTrack(track: MediaStreamTrack)\n    fun removeTrack(track: MediaStreamTrack)\n
    fun clone(): MediaStream\n}\n\n/**\n * Exposes the JavaScript
[MediaStreamTrack](https://developer.mozilla.org/en/docs/Web/API/MediaStreamTrack) to Kotlin\n */\npublic
external abstract class MediaStreamTrack : EventTarget {\n    open val kind: String\n    open val id: String\n    open
val label: String\n    open var enabled: Boolean\n    open val muted: Boolean\n    open var onmute: ((Event) ->
dynamic)?\n    open var onunmute: ((Event) -> dynamic)?\n    open val readyState: MediaStreamTrackState\n
open var onended: ((Event) -> dynamic)?\n    open var onoverconstrained: ((Event) -> dynamic)?\n    fun clone():
MediaStreamTrack\n    fun stop()\n    fun getCapabilities(): MediaTrackCapabilities\n    fun getConstraints():
MediaTrackConstraints\n    fun getSettings(): MediaTrackSettings\n    fun applyConstraints(constraints:
MediaTrackConstraints = definedExternally): Promise<Unit>\n}\n\n/**\n * Exposes the JavaScript
[MediaTrackSupportedConstraints](https://developer.mozilla.org/en/docs/Web/API/MediaTrackSupportedConstrain
ts) to Kotlin\n */\npublic external interface MediaTrackSupportedConstraints {\n    var width: Boolean? /* = true
*/\n    get() = definedExternally\n    set(value) = definedExternally\n    var height: Boolean? /* = true */\n
get() = definedExternally\n    set(value) = definedExternally\n    var aspectRatio: Boolean? /* = true */\n    get()
= definedExternally\n    set(value) = definedExternally\n    var frameRate: Boolean? /* = true */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var facingMode: Boolean? /* = true */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var resizeMode: Boolean? /* = true */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var volume: Boolean? /* = true */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var sampleRate: Boolean? /* = true */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var sampleSize: Boolean? /* = true */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var echoCancellation: Boolean? /* = true */\n    get()
= definedExternally\n    set(value) = definedExternally\n    var autoGainControl: Boolean? /* = true */\n    get()
= definedExternally\n    set(value) = definedExternally\n    var noiseSuppression: Boolean? /* = true */\n
get() = definedExternally\n    set(value) = definedExternally\n    var latency: Boolean? /* = true */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var channelCount: Boolean? /* = true */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var deviceId: Boolean? /* = true */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var groupId: Boolean? /* = true */\n    get() =
definedExternally\n    set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun
MediaTrackSupportedConstraints(width: Boolean? = true, height: Boolean? = true, aspectRatio: Boolean? = true,
frameRate: Boolean? = true, facingMode: Boolean? = true, resizeMode: Boolean? = true, volume: Boolean? = true,
sampleRate: Boolean? = true, sampleSize: Boolean? = true, echoCancellation: Boolean? = true, autoGainControl:
Boolean? = true, noiseSuppression: Boolean? = true, latency: Boolean? = true, channelCount: Boolean? = true,
deviceId: Boolean? = true, groupId: Boolean? = true): MediaTrackSupportedConstraints {\n    val o = js("{}")\n
o["width"] = width\n    o["height"] = height\n    o["aspectRatio"] = aspectRatio\n    o["frameRate"] =
frameRate\n    o["facingMode"] = facingMode\n    o["resizeMode"] = resizeMode\n    o["volume"] = volume\n
o["sampleRate"] = sampleRate\n    o["sampleSize"] = sampleSize\n    o["echoCancellation"] =

```

```

echoCancellation\n  o["autoGainControl"] = autoGainControl\n  o["noiseSuppression"] = noiseSuppression\n
o["latency"] = latency\n  o["channelCount"] = channelCount\n  o["deviceId"] = deviceId\n  o["groupId"] =
groupId\n  return o\n}\n\npublic external interface MediaTrackCapabilities {\n  var width: ULongRange?\n
get() = definedExternally\n  set(value) = definedExternally\n  var height: ULongRange?\n  get() =
definedExternally\n  set(value) = definedExternally\n  var aspectRatio: DoubleRange?\n  get() =
definedExternally\n  set(value) = definedExternally\n  var frameRate: DoubleRange?\n  get() =
definedExternally\n  set(value) = definedExternally\n  var facingMode: Array<String>?\n  get() =
definedExternally\n  set(value) = definedExternally\n  var resizeMode: Array<String>?\n  get() =
definedExternally\n  set(value) = definedExternally\n  var volume: DoubleRange?\n  get() =
definedExternally\n  set(value) = definedExternally\n  var sampleRate: ULongRange?\n  get() =
definedExternally\n  set(value) = definedExternally\n  var sampleSize: ULongRange?\n  get() =
definedExternally\n  set(value) = definedExternally\n  var echoCancellation: Array<Boolean>?\n  get() =
definedExternally\n  set(value) = definedExternally\n  var autoGainControl: Array<Boolean>?\n  get() =
definedExternally\n  set(value) = definedExternally\n  var noiseSuppression: Array<Boolean>?\n  get() =
definedExternally\n  set(value) = definedExternally\n  var latency: DoubleRange?\n  get() =
definedExternally\n  set(value) = definedExternally\n  var channelCount: ULongRange?\n  get() =
definedExternally\n  set(value) = definedExternally\n  var deviceId: String?\n  get() = definedExternally\n
set(value) = definedExternally\n  var groupId: String?\n  get() = definedExternally\n  set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun MediaTrackCapabilities(width:
ULongRange? = undefined, height: ULongRange? = undefined, aspectRatio: DoubleRange? = undefined,
frameRate: DoubleRange? = undefined, facingMode: Array<String>? = undefined, resizeMode: Array<String>? =
undefined, volume: DoubleRange? = undefined, sampleRate: ULongRange? = undefined, sampleSize:
ULongRange? = undefined, echoCancellation: Array<Boolean>? = undefined, autoGainControl: Array<Boolean>?
= undefined, noiseSuppression: Array<Boolean>? = undefined, latency: DoubleRange? = undefined, channelCount:
ULongRange? = undefined, deviceId: String? = undefined, groupId: String? = undefined): MediaTrackCapabilities
{\n  val o = js("{}")\n  o["width"] = width\n  o["height"] = height\n  o["aspectRatio"] = aspectRatio\n
o["frameRate"] = frameRate\n  o["facingMode"] = facingMode\n  o["resizeMode"] = resizeMode\n
o["volume"] = volume\n  o["sampleRate"] = sampleRate\n  o["sampleSize"] = sampleSize\n
o["echoCancellation"] = echoCancellation\n  o["autoGainControl"] = autoGainControl\n
o["noiseSuppression"] = noiseSuppression\n  o["latency"] = latency\n  o["channelCount"] = channelCount\n
o["deviceId"] = deviceId\n  o["groupId"] = groupId\n  return o\n}\n\n/**\n * Exposes the JavaScript
[MediaTrackConstraints](https://developer.mozilla.org/en/docs/Web/API/MediaTrackConstraints) to Kotlin\n
*/\n\npublic external interface MediaTrackConstraints : MediaTrackConstraintSet {\n  var advanced:
Array<MediaTrackConstraintSet>?\n  get() = definedExternally\n  set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun MediaTrackConstraints(advanced:
Array<MediaTrackConstraintSet>? = undefined, width: dynamic = undefined, height: dynamic = undefined,
aspectRatio: dynamic = undefined, frameRate: dynamic = undefined, facingMode: dynamic = undefined,
resizeMode: dynamic = undefined, volume: dynamic = undefined, sampleRate: dynamic = undefined, sampleSize:
dynamic = undefined, echoCancellation: dynamic = undefined, autoGainControl: dynamic = undefined,
noiseSuppression: dynamic = undefined, latency: dynamic = undefined, channelCount: dynamic = undefined,
deviceId: dynamic = undefined, groupId: dynamic = undefined): MediaTrackConstraints {\n  val o = js("{}")\n
o["advanced"] = advanced\n  o["width"] = width\n  o["height"] = height\n  o["aspectRatio"] =
aspectRatio\n  o["frameRate"] = frameRate\n  o["facingMode"] = facingMode\n  o["resizeMode"] =
resizeMode\n  o["volume"] = volume\n  o["sampleRate"] = sampleRate\n  o["sampleSize"] = sampleSize\n
o["echoCancellation"] = echoCancellation\n  o["autoGainControl"] = autoGainControl\n
o["noiseSuppression"] = noiseSuppression\n  o["latency"] = latency\n  o["channelCount"] = channelCount\n

```

```

o["deviceId"] = deviceId\n  o["groupId"] = groupId\n  return o\n}\n\npublic external interface
MediaTrackConstraintSet {\n  var width: dynamic\n    get() = definedExternally\n    set(value) =
definedExternally\n  var height: dynamic\n    get() = definedExternally\n    set(value) = definedExternally\n
var aspectRatio: dynamic\n    get() = definedExternally\n    set(value) = definedExternally\n  var frameRate:
dynamic\n    get() = definedExternally\n    set(value) = definedExternally\n  var facingMode: dynamic\n
get() = definedExternally\n    set(value) = definedExternally\n  var resizeMode: dynamic\n    get() =
definedExternally\n    set(value) = definedExternally\n  var volume: dynamic\n    get() = definedExternally\n
set(value) = definedExternally\n  var sampleRate: dynamic\n    get() = definedExternally\n    set(value) =
definedExternally\n  var sampleSize: dynamic\n    get() = definedExternally\n    set(value) =
definedExternally\n  var echoCancellation: dynamic\n    get() = definedExternally\n    set(value) =
definedExternally\n  var autoGainControl: dynamic\n    get() = definedExternally\n    set(value) =
definedExternally\n  var noiseSuppression: dynamic\n    get() = definedExternally\n    set(value) =
definedExternally\n  var latency: dynamic\n    get() = definedExternally\n    set(value) = definedExternally\n
var channelCount: dynamic\n    get() = definedExternally\n    set(value) = definedExternally\n  var deviceId:
dynamic\n    get() = definedExternally\n    set(value) = definedExternally\n  var groupId: dynamic\n    get()
= definedExternally\n    set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun MediaTrackConstraintSet(width:
dynamic = undefined, height: dynamic = undefined, aspectRatio: dynamic = undefined, frameRate: dynamic =
undefined, facingMode: dynamic = undefined, resizeMode: dynamic = undefined, volume: dynamic = undefined,
sampleRate: dynamic = undefined, sampleSize: dynamic = undefined, echoCancellation: dynamic = undefined,
autoGainControl: dynamic = undefined, noiseSuppression: dynamic = undefined, latency: dynamic = undefined,
channelCount: dynamic = undefined, deviceId: dynamic = undefined, groupId: dynamic = undefined):
MediaTrackConstraintSet {\n  val o = js("{}")\n  o["width"] = width\n  o["height"] = height\n
o["aspectRatio"] = aspectRatio\n  o["frameRate"] = frameRate\n  o["facingMode"] = facingMode\n
o["resizeMode"] = resizeMode\n  o["volume"] = volume\n  o["sampleRate"] = sampleRate\n
o["sampleSize"] = sampleSize\n  o["echoCancellation"] = echoCancellation\n  o["autoGainControl"] =
autoGainControl\n  o["noiseSuppression"] = noiseSuppression\n  o["latency"] = latency\n
o["channelCount"] = channelCount\n  o["deviceId"] = deviceId\n  o["groupId"] = groupId\n  return
o\n}\n\n/**\n * Exposes the JavaScript
[MediaTrackSettings](https://developer.mozilla.org/en/docs/Web/API/MediaTrackSettings) to Kotlin\n *\npublic
external interface MediaTrackSettings {\n  var width: Int?\n    get() = definedExternally\n    set(value) =
definedExternally\n  var height: Int?\n    get() = definedExternally\n    set(value) = definedExternally\n
var aspectRatio: Double?\n    get() = definedExternally\n    set(value) = definedExternally\n  var frameRate:
Double?\n    get() = definedExternally\n    set(value) = definedExternally\n  var facingMode: String?\n
get() = definedExternally\n    set(value) = definedExternally\n  var resizeMode: String?\n    get() =
definedExternally\n    set(value) = definedExternally\n  var volume: Double?\n    get() = definedExternally\n
set(value) = definedExternally\n  var sampleRate: Int?\n    get() = definedExternally\n    set(value) =
definedExternally\n  var sampleSize: Int?\n    get() = definedExternally\n    set(value) = definedExternally\n
var echoCancellation: Boolean?\n    get() = definedExternally\n    set(value) = definedExternally\n  var
autoGainControl: Boolean?\n    get() = definedExternally\n    set(value) = definedExternally\n  var
noiseSuppression: Boolean?\n    get() = definedExternally\n    set(value) = definedExternally\n  var latency:
Double?\n    get() = definedExternally\n    set(value) = definedExternally\n  var channelCount: Int?\n
get() = definedExternally\n    set(value) = definedExternally\n  var deviceId: String?\n    get() =
definedExternally\n    set(value) = definedExternally\n  var groupId: String?\n    get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun MediaTrackSettings(width: Int? =
undefined, height: Int? = undefined, aspectRatio: Double? = undefined, frameRate: Double? = undefined,
facingMode: String? = undefined, resizeMode: String? = undefined, volume: Double? = undefined, sampleRate: Int?

```

```

= undefined, sampleSize: Int? = undefined, echoCancellation: Boolean? = undefined, autoGainControl: Boolean? =
undefined, noiseSuppression: Boolean? = undefined, latency: Double? = undefined, channelCount: Int? = undefined,
deviceId: String? = undefined, groupId: String? = undefined): MediaTrackSettings {\n  val o = js("{}")\n
o["width"] = width\n  o["height"] = height\n  o["aspectRatio"] = aspectRatio\n  o["frameRate"] =
frameRate\n  o["facingMode"] = facingMode\n  o["resizeMode"] = resizeMode\n  o["volume"] = volume\n
o["sampleRate"] = sampleRate\n  o["sampleSize"] = sampleSize\n  o["echoCancellation"] =
echoCancellation\n  o["autoGainControl"] = autoGainControl\n  o["noiseSuppression"] = noiseSuppression\n
o["latency"] = latency\n  o["channelCount"] = channelCount\n  o["deviceId"] = deviceId\n  o["groupId"] =
groupId\n  return o\n}\n\n**\n * Exposes the JavaScript
[MediaStreamTrackEvent](https://developer.mozilla.org/en/docs/Web/API/MediaStreamTrackEvent) to Kotlin\n
*\npublic external open class MediaStreamTrackEvent(type: String, eventInitDict: MediaStreamTrackEventInit) :
Event {\n  open val track: MediaStreamTrack\n\n  companion object {\n    val NONE: Short\n    val
CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val BUBBLING_PHASE: Short\n
}\n}\n\npublic external interface MediaStreamTrackEventInit : EventInit {\n  var track:
MediaStreamTrack?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun MediaStreamTrackEventInit(track:
MediaStreamTrack?, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false):
MediaStreamTrackEventInit {\n  val o = js("{}")\n  o["track"] = track\n  o["bubbles"] = bubbles\n
o["cancelable"] = cancelable\n  o["composed"] = composed\n  return o\n}\n\npublic external open class
OverconstrainedErrorEvent(type: String, eventInitDict: OverconstrainedErrorEventInit) : Event {\n  open val error:
dynamic\n\n  companion object {\n    val NONE: Short\n    val CAPTURING_PHASE: Short\n    val
AT_TARGET: Short\n    val BUBBLING_PHASE: Short\n  }\n}\n\npublic external interface
OverconstrainedErrorEventInit : EventInit {\n  var error: dynamic /* = null */\n  get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun OverconstrainedErrorEventInit(error:
dynamic = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false):
OverconstrainedErrorEventInit {\n  val o = js("{}")\n  o["error"] = error\n  o["bubbles"] = bubbles\n
o["cancelable"] = cancelable\n  o["composed"] = composed\n  return o\n}\n\n**\n * Exposes the JavaScript
[MediaDevices](https://developer.mozilla.org/en/docs/Web/API/MediaDevices) to Kotlin\n
*\npublic external
abstract class MediaDevices : EventTarget {\n  open var ondevicechange: ((Event) -> dynamic)?\n  fun
enumerateDevices(): Promise<Array<MediaDeviceInfo>>\n  fun getSupportedConstraints():
MediaTrackSupportedConstraints\n  fun getUserMedia(constraints: MediaStreamConstraints = definedExternally):
Promise<MediaStream>\n}\n\n**\n * Exposes the JavaScript
[MediaDeviceInfo](https://developer.mozilla.org/en/docs/Web/API/MediaDeviceInfo) to Kotlin\n
*\npublic
external abstract class MediaDeviceInfo {\n  open val deviceId: String\n  open val kind: MediaDeviceKind\n
open val label: String\n  open val groupId: String\n  fun toJSON(): dynamic\n}\n\npublic external abstract class
InputDeviceInfo : MediaDeviceInfo {\n  fun getCapabilities(): MediaTrackCapabilities\n}\n\n**\n * Exposes the
JavaScript [MediaStreamConstraints](https://developer.mozilla.org/en/docs/Web/API/MediaStreamConstraints) to
Kotlin\n
*\npublic external interface MediaStreamConstraints {\n  var video: dynamic /* = false */\n  get() =
definedExternally\n  set(value) = definedExternally\n  var audio: dynamic /* = false */\n  get() =
definedExternally\n  set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun MediaStreamConstraints(video:
dynamic = false, audio: dynamic = false): MediaStreamConstraints {\n  val o = js("{}")\n  o["video"] =
video\n  o["audio"] = audio\n  return o\n}\n\npublic external interface ConstrainingPattern {\n  var
onoverconstrained: ((Event) -> dynamic)?\n  get() = definedExternally\n  set(value) = definedExternally\n
fun getCapabilities(): Capabilities\n  fun getConstraints(): Constraints\n  fun getSettings(): Settings\n  fun
applyConstraints(constraints: Constraints = definedExternally): Promise<Unit>\n}\n\n**\n * Exposes the
JavaScript [DoubleRange](https://developer.mozilla.org/en/docs/Web/API/DoubleRange) to Kotlin\n
*\npublic

```



```

external interface DoubleRange {
    var max: Double?
    get() = definedExternally
    set(value) = definedExternally
    var min: Double?
    get() = definedExternally
    set(value) = definedExternally
}
@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun DoubleRange(max: Double? = undefined, min: Double? = undefined): DoubleRange {
    val o = js("{}")
    o["max"] = max
    o["min"] = min
    return o
}
public external interface ConstrainDoubleRange : DoubleRange {
    var exact: Double?
    get() = definedExternally
    set(value) = definedExternally
    var ideal: Double?
    get() = definedExternally
    set(value) = definedExternally
}
@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun ConstrainDoubleRange(exact: Double? = undefined, ideal: Double? = undefined, max: Double? = undefined, min: Double? = undefined): ConstrainDoubleRange {
    val o = js("{}")
    o["exact"] = exact
    o["ideal"] = ideal
    o["max"] = max
    o["min"] = min
    return o
}
public external interface ULongRange {
    var max: Int?
    get() = definedExternally
    set(value) = definedExternally
    var min: Int?
    get() = definedExternally
    set(value) = definedExternally
}
@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun ULongRange(max: Int? = undefined, min: Int? = undefined): ULongRange {
    val o = js("{}")
    o["max"] = max
    o["min"] = min
    return o
}
public external interface ConstrainULongRange : ULongRange {
    var exact: Int?
    get() = definedExternally
    set(value) = definedExternally
    var ideal: Int?
    get() = definedExternally
    set(value) = definedExternally
}
@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun ConstrainULongRange(exact: Int? = undefined, ideal: Int? = undefined, max: Int? = undefined, min: Int? = undefined): ConstrainULongRange {
    val o = js("{}")
    o["exact"] = exact
    o["ideal"] = ideal
    o["max"] = max
    o["min"] = min
    return o
}
}
** Exposes the JavaScript
[ConstrainBooleanParameters](https://developer.mozilla.org/en/docs/Web/API/ConstrainBooleanParameters) to
Kotlin
public external interface ConstrainBooleanParameters {
    var exact: Boolean?
    get() = definedExternally
    set(value) = definedExternally
    var ideal: Boolean?
    get() = definedExternally
    set(value) = definedExternally
}
@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun ConstrainBooleanParameters(exact: Boolean? = undefined, ideal: Boolean? = undefined): ConstrainBooleanParameters {
    val o = js("{}")
    o["exact"] = exact
    o["ideal"] = ideal
    return o
}
}
** Exposes the JavaScript
[ConstrainDOMStringParameters](https://developer.mozilla.org/en/docs/Web/API/ConstrainDOMStringParameters)
to Kotlin
public external interface ConstrainDOMStringParameters {
    var exact: dynamic
    get() = definedExternally
    set(value) = definedExternally
    var ideal: dynamic
    get() = definedExternally
    set(value) = definedExternally
}
@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun ConstrainDOMStringParameters(exact: dynamic = undefined, ideal: dynamic = undefined): ConstrainDOMStringParameters {
    val o = js("{}")
    o["exact"] = exact
    o["ideal"] = ideal
    return o
}
}
public external interface Capabilities
@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun Capabilities(): Capabilities {
    val o = js("{}")
    return o
}
}
public external interface Settings
@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun Settings(): Settings {
    val o = js("{}")
    return o
}
}
public external interface ConstraintSet
@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun ConstraintSet(): ConstraintSet {
    val o = js("{}")
    return o
}
}
public external interface Constraints : ConstraintSet {
    var advanced: Array<ConstraintSet>?
    get() = definedExternally
    set(value) = definedExternally
}
@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun Constraints(advanced: Array<ConstraintSet>? = undefined): Constraints {
    val o = js("{}")
    o["advanced"] = advanced
}

```

```

return o\n}\n\n/* please, don't implement this interface!
*\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface MediaStreamTrackState {\n    companion object\n}\n\npublic inline val
MediaStreamTrackState.Companion.LIVE: MediaStreamTrackState get() =
"live".asDynamic().unsafeCast<MediaStreamTrackState>()\n\npublic inline val
MediaStreamTrackState.Companion.ENDED: MediaStreamTrackState get() =
"ended".asDynamic().unsafeCast<MediaStreamTrackState>()\n\n/* please, don't implement this interface!
*\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface VideoFacingModeEnum {\n    companion object\n}\n\npublic inline val
VideoFacingModeEnum.Companion.USER: VideoFacingModeEnum get() =
"user".asDynamic().unsafeCast<VideoFacingModeEnum>()\n\npublic inline val
VideoFacingModeEnum.Companion.ENVIRONMENT: VideoFacingModeEnum get() =
"environment".asDynamic().unsafeCast<VideoFacingModeEnum>()\n\npublic inline val
VideoFacingModeEnum.Companion.LEFT: VideoFacingModeEnum get() =
"left".asDynamic().unsafeCast<VideoFacingModeEnum>()\n\npublic inline val
VideoFacingModeEnum.Companion.RIGHT: VideoFacingModeEnum get() =
"right".asDynamic().unsafeCast<VideoFacingModeEnum>()\n\n/* please, don't implement this interface!
*\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface VideoResizeModeEnum {\n    companion object\n}\n\npublic inline val
VideoResizeModeEnum.Companion.NONE: VideoResizeModeEnum get() =
"none".asDynamic().unsafeCast<VideoResizeModeEnum>()\n\npublic inline val
VideoResizeModeEnum.Companion.CROP_AND_SCALE: VideoResizeModeEnum get() = "crop-and-
scale".asDynamic().unsafeCast<VideoResizeModeEnum>()\n\n/* please, don't implement this interface!
*\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface MediaDeviceKind {\n    companion object\n}\n\npublic inline val
MediaDeviceKind.Companion.AUDIOINPUT: MediaDeviceKind get() =
"audioinput".asDynamic().unsafeCast<MediaDeviceKind>()\n\npublic inline val
MediaDeviceKind.Companion.AUDIOOUTPUT: MediaDeviceKind get() =
"audiooutput".asDynamic().unsafeCast<MediaDeviceKind>()\n\npublic inline val
MediaDeviceKind.Companion.VIDEOINPUT: MediaDeviceKind get() =
"videoinput".asDynamic().unsafeCast<MediaDeviceKind>())"/*\n * Copyright 2010-2021 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT
EDIT!\n\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.dom.mediasource\n\nimport
kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\n\n/**\n * Exposes
the JavaScript [MediaSource](https://developer.mozilla.org/en/docs/Web/API/MediaSource) to Kotlin\n */\n\npublic
external open class MediaSource : EventTarget, MediaProvider {\n    open val sourceBuffers: SourceBufferList\n
open val activeSourceBuffers: SourceBufferList\n    open val readyState: ReadyState\n    var duration: Double\n
var onsourceopen: ((Event) -> dynamic)?\n    var onsourceended: ((Event) -> dynamic)?\n    var onsourceclose:
((Event) -> dynamic)?\n    fun addSourceBuffer(type: String): SourceBuffer\n    fun
removeSourceBuffer(sourceBuffer: SourceBuffer)\n    fun endOfStream(error: EndOfStreamError =
definedExternally)\n    fun setLiveSeekableRange(start: Double, end: Double)\n    fun clearLiveSeekableRange()\n\n
companion object {\n    fun isTypeSupported(type: String): Boolean\n    }\n}\n\n/**\n * Exposes the JavaScript
[SourceBuffer](https://developer.mozilla.org/en/docs/Web/API/SourceBuffer) to Kotlin\n */\n\npublic external
abstract class SourceBuffer : EventTarget {\n    open var mode: AppendMode\n    open val updating: Boolean\n
open val buffered: TimeRanges\n    open val timestampOffset: Double\n    open val audioTracks: AudioTrackList\n
open val videoTracks: VideoTrackList\n    open val textTracks: TextTrackList\n    open var appendWindowStart:
Double\n    open var appendWindowEnd: Double\n    open var onupdatestart: ((Event) -> dynamic)?\n    open var

```

```

onupdate: ((Event) -> dynamic)?\n  open var onupdateend: ((Event) -> dynamic)?\n  open var onerror: ((Event) ->
dynamic)?\n  open var onabort: ((Event) -> dynamic)?\n  fun appendBuffer(data: dynamic)\n  fun abort()\n  fun
remove(start: Double, end: Double)\n}\n\n/**\n * Exposes the JavaScript
[SourceBufferList](https://developer.mozilla.org/en/docs/Web/API/SourceBufferList) to Kotlin\n */\npublic
external abstract class SourceBufferList : EventTarget {\n  open val length: Int\n  open var onaddsourcebuffer:
((Event) -> dynamic)?\n  open var onremovesourcebuffer: ((Event) ->
dynamic)?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SourceBufferList.get(index:
Int): SourceBuffer? = asDynamic()[index]\n\n/* please, don't implement this interface!
*\n */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface ReadyState {\n  companion object\n}\n\npublic inline val ReadyState.Companion.CLOSED: ReadyState
get() = \"closed\".asDynamic().unsafeCast<ReadyState>()\n\npublic inline val ReadyState.Companion.OPEN:
ReadyState get() = \"open\".asDynamic().unsafeCast<ReadyState>()\n\npublic inline val
ReadyState.Companion.ENDED: ReadyState get() = \"ended\".asDynamic().unsafeCast<ReadyState>()\n\n/*
please, don't implement this interface!
*\n */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface EndOfStreamError {\n  companion object\n}\n\npublic inline val
EndOfStreamError.Companion.NETWORK: EndOfStreamError get() =
\"network\".asDynamic().unsafeCast<EndOfStreamError>()\n\npublic inline val
EndOfStreamError.Companion.DECODE: EndOfStreamError get() =
\"decode\".asDynamic().unsafeCast<EndOfStreamError>()\n\n/* please, don't implement this interface!
*\n */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface AppendMode {\n  companion object\n}\n\npublic inline val AppendMode.Companion.SEGMENTS:
AppendMode get() = \"segments\".asDynamic().unsafeCast<AppendMode>()\n\npublic inline val
AppendMode.Companion.SEQUENCE: AppendMode get() =
\"sequence\".asDynamic().unsafeCast<AppendMode>())\n\n/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\n/* NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n */\n\n//
See github.com/kotlin/dukat for details\n\npackage org.w3c.dom.pointerevents\n\nimport kotlin.js.*\nimport
org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\n\npublic external interface
PointerEventInit : MouseEventInit {\n  var pointerId: Int? /* = 0 */\n  get() = definedExternally\n  set(value) = definedExternally\n  var width: Double? /* = 1.0 */\n  get() = definedExternally\n  set(value) =
definedExternally\n  var height: Double? /* = 1.0 */\n  get() = definedExternally\n  set(value) =
definedExternally\n  var pressure: Float? /* = 0f */\n  get() = definedExternally\n  set(value) =
definedExternally\n  var tangentialPressure: Float? /* = 0f */\n  get() = definedExternally\n  set(value) =
definedExternally\n  var tiltX: Int? /* = 0 */\n  get() = definedExternally\n  set(value) = definedExternally\n
  var tiltY: Int? /* = 0 */\n  get() = definedExternally\n  set(value) = definedExternally\n  var twist: Int? /* =
0 */\n  get() = definedExternally\n  set(value) = definedExternally\n  var pointerType: String? /* = \"\" */\n
  get() = definedExternally\n  set(value) = definedExternally\n  var isPrimary: Boolean? /* = false */\n
  get() = definedExternally\n  set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun PointerEventInit(pointerId: Int? = 0,
width: Double? = 1.0, height: Double? = 1.0, pressure: Float? = 0f, tangentialPressure: Float? = 0f, tiltX: Int? = 0,
tiltY: Int? = 0, twist: Int? = 0, pointerType: String? = \"\", isPrimary: Boolean? = false, screenX: Int? = 0, screenY:
Int? = 0, clientX: Int? = 0, clientY: Int? = 0, button: Short? = 0, buttons: Short? = 0, relatedTarget: EventTarget? =
null, region: String? = null, ctrlKey: Boolean? = false, shiftKey: Boolean? = false, altKey: Boolean? = false,
metaKey: Boolean? = false, modifierAltGraph: Boolean? = false, modifierCapsLock: Boolean? = false, modifierFn:
Boolean? = false, modifierFnLock: Boolean? = false, modifierHyper: Boolean? = false, modifierNumLock:
Boolean? = false, modifierScrollLock: Boolean? = false, modifierSuper: Boolean? = false, modifierSymbol:

```

```

Boolean? = false, modifierSymbolLock: Boolean? = false, view: Window? = null, detail: Int? = 0, bubbles:
Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): PointerEventInit {\n  val o =
js("{}")\n  o["pointerId"] = pointerId\n  o["width"] = width\n  o["height"] = height\n  o["pressure"] =
pressure\n  o["tangentialPressure"] = tangentialPressure\n  o["tiltX"] = tiltX\n  o["tiltY"] = tiltY\n
o["twist"] = twist\n  o["pointerType"] = pointerType\n  o["isPrimary"] = isPrimary\n  o["screenX"] =
screenX\n  o["screenY"] = screenY\n  o["clientX"] = clientX\n  o["clientY"] = clientY\n  o["button"] =
button\n  o["buttons"] = buttons\n  o["relatedTarget"] = relatedTarget\n  o["region"] = region\n
o["ctrlKey"] = ctrlKey\n  o["shiftKey"] = shiftKey\n  o["altKey"] = altKey\n  o["metaKey"] = metaKey\n
o["modifierAltGraph"] = modifierAltGraph\n  o["modifierCapsLock"] = modifierCapsLock\n
o["modifierFn"] = modifierFn\n  o["modifierFnLock"] = modifierFnLock\n  o["modifierHyper"] =
modifierHyper\n  o["modifierNumLock"] = modifierNumLock\n  o["modifierScrollLock"] =
modifierScrollLock\n  o["modifierSuper"] = modifierSuper\n  o["modifierSymbol"] = modifierSymbol\n
o["modifierSymbolLock"] = modifierSymbolLock\n  o["view"] = view\n  o["detail"] = detail\n
o["bubbles"] = bubbles\n  o["cancelable"] = cancelable\n  o["composed"] = composed\n  return
o}\n\n/**\n * Exposes the JavaScript

```

```

[PointerEvent](https://developer.mozilla.org/en/docs/Web/API/PointerEvent) to Kotlin\n\npublic external open
class PointerEvent(type: String, eventInitDict: PointerEventInit = definedExternally) : MouseEvent {\n  open val
pointerId: Int\n  open val width: Double\n  open val height: Double\n  open val pressure: Float\n  open val
tangentialPressure: Float\n  open val tiltX: Int\n  open val tiltY: Int\n  open val twist: Int\n  open val
pointerType: String\n  open val isPrimary: Boolean\n\n  companion object {\n    val NONE: Short\n    val
CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val BUBBLING_PHASE: Short\n  }\n}"/**\n
* Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n\n// NOTE: THIS
FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage
org.w3c.dom.svg\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\nimport
org.w3c.dom.css.*\n\n/**\n * Exposes the JavaScript

```

```

[SVGElement](https://developer.mozilla.org/en/docs/Web/API/SVGElement) to Kotlin\n\npublic external
abstract class SVGElement : Element, ElementCSSInlineStyle, GlobalEventHandlers, SVGElementInstance {\n
open val dataset: DOMStringMap\n  open val ownerSVGElement: SVGSVGElement?\n  open val
viewportElement: SVGElement?\n  open var tabIndex: Int\n  fun focus()\n  fun blur()\n\n  companion object
{\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val
CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val
ENTITY_NODE: Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE:
Short\n    val DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n\n  public external interface
SVGBoundingBoxOptions {\n    var fill: Boolean? /* = true */\n    get() = definedExternally\n    set(value) =
definedExternally\n    var stroke: Boolean? /* = false */\n    get() = definedExternally\n    set(value) =
definedExternally\n    var markers: Boolean? /* = false */\n    get() = definedExternally\n    set(value) =
definedExternally\n    var clipped: Boolean? /* = false */\n    get() = definedExternally\n    set(value) =
definedExternally\n  }\n\n  @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n  @kotlin.internal.InlineOnly\n  public inline fun SVGBoundingBoxOptions(fill:
Boolean? = true, stroke: Boolean? = false, markers: Boolean? = false, clipped: Boolean? = false):
SVGBoundingBoxOptions {\n    val o = js("{}")\n    o["fill"] = fill\n    o["stroke"] = stroke\n    o["markers"]
= markers\n    o["clipped"] = clipped\n    return o\n  }\n\n  /**\n * Exposes the JavaScript

```

```

[SVGGraphicsElement](https://developer.mozilla.org/en/docs/Web/API/SVGGraphicsElement) to Kotlin\n
*\npublic external abstract class SVGGraphicsElement : SVGElement, SVGTests {\n  open val transform:\n  SVGAnimatedTransformList\n  fun getBBox(options: SVGBoundingBoxOptions = definedExternally):\n  DOMRect\n  fun getCTM(): DOMMatrix?\n  fun getScreenCTM(): DOMMatrix?\n\n  companion object {\n  val ELEMENT_NODE: Short\n  val ATTRIBUTE_NODE: Short\n  val TEXT_NODE: Short\n  val\n  CDATA_SECTION_NODE: Short\n  val ENTITY_REFERENCE_NODE: Short\n  val ENTITY_NODE:\n  Short\n  val PROCESSING_INSTRUCTION_NODE: Short\n  val COMMENT_NODE: Short\n  val\n  DOCUMENT_NODE: Short\n  val DOCUMENT_TYPE_NODE: Short\n  val\n  DOCUMENT_FRAGMENT_NODE: Short\n  val NOTATION_NODE: Short\n  val\n  DOCUMENT_POSITION_DISCONNECTED: Short\n  val DOCUMENT_POSITION_PRECEDING: Short\n  val\n  DOCUMENT_POSITION_FOLLOWING: Short\n  val DOCUMENT_POSITION_CONTAINS: Short\n  val\n  DOCUMENT_POSITION_CONTAINED_BY: Short\n  val\n  DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n}\n\n/**\n * Exposes the JavaScript\n [SVGGeometryElement](https://developer.mozilla.org/en/docs/Web/API/SVGGeometryElement) to Kotlin\n *\npublic external abstract class SVGGeometryElement : SVGGraphicsElement {\n  open val pathLength:\n  SVGAnimatedNumber\n  fun isPointInFill(point: DOMPoint): Boolean\n  fun isPointInStroke(point: DOMPoint):\n  Boolean\n  fun getTotalLength(): Float\n  fun getPointAtLength(distance: Float): DOMPoint\n\n  companion\n  object {\n  val ELEMENT_NODE: Short\n  val ATTRIBUTE_NODE: Short\n  val TEXT_NODE:\n  Short\n  val CDATA_SECTION_NODE: Short\n  val ENTITY_REFERENCE_NODE: Short\n  val\n  ENTITY_NODE: Short\n  val PROCESSING_INSTRUCTION_NODE: Short\n  val COMMENT_NODE:\n  Short\n  val DOCUMENT_NODE: Short\n  val DOCUMENT_TYPE_NODE: Short\n  val\n  DOCUMENT_FRAGMENT_NODE: Short\n  val NOTATION_NODE: Short\n  val\n  DOCUMENT_POSITION_DISCONNECTED: Short\n  val DOCUMENT_POSITION_PRECEDING: Short\n  val\n  DOCUMENT_POSITION_FOLLOWING: Short\n  val DOCUMENT_POSITION_CONTAINS: Short\n  val\n  DOCUMENT_POSITION_CONTAINED_BY: Short\n  val\n  DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n}\n\n/**\n * Exposes the JavaScript\n [SVGNumber](https://developer.mozilla.org/en/docs/Web/API/SVGNumber) to Kotlin\n *\npublic external\n  abstract class SVGNumber {\n  open var value: Float\n}\n\n\n/**\n * Exposes the JavaScript\n [SVGLength](https://developer.mozilla.org/en/docs/Web/API/SVGLength) to Kotlin\n *\npublic external abstract\n  class SVGLength {\n  open val unitType: Short\n  open var value: Float\n  open var valueInSpecifiedUnits:\n  Float\n  open var valueAsString: String\n  fun newValueSpecifiedUnits(unitType: Short, valueInSpecifiedUnits:\n  Float)\n  fun convertToSpecifiedUnits(unitType: Short)\n\n  companion object {\n  val\n  SVG_LENGTHTYPE_UNKNOWN: Short\n  val SVG_LENGTHTYPE_NUMBER: Short\n  val\n  SVG_LENGTHTYPE_PERCENTAGE: Short\n  val SVG_LENGTHTYPE_EMS: Short\n  val\n  SVG_LENGTHTYPE_EXS: Short\n  val SVG_LENGTHTYPE_PX: Short\n  val\n  SVG_LENGTHTYPE_CM: Short\n  val SVG_LENGTHTYPE_MM: Short\n  val\n  SVG_LENGTHTYPE_IN: Short\n  val SVG_LENGTHTYPE_PT: Short\n  val SVG_LENGTHTYPE_PC:\n  Short\n  }\n}\n\n\n/**\n * Exposes the JavaScript\n [SVGAngle](https://developer.mozilla.org/en/docs/Web/API/SVGAngle) to Kotlin\n *\npublic external abstract\n  class SVGAngle {\n  open val unitType: Short\n  open var value: Float\n  open var valueInSpecifiedUnits:\n  Float\n  open var valueAsString: String\n  fun newValueSpecifiedUnits(unitType: Short, valueInSpecifiedUnits:\n  Float)\n  fun convertToSpecifiedUnits(unitType: Short)\n\n  companion object {\n  val\n  SVG_ANGLETYPE_UNKNOWN: Short\n  val SVG_ANGLETYPE_UNSPECIFIED: Short\n  val\n  SVG_ANGLETYPE_DEG: Short\n  val SVG_ANGLETYPE_RAD: Short\n  val\n  SVG_ANGLETYPE_GRAD: Short\n  }\n}\n\n\npublic external abstract class SVGNameList {\n  open val length:\n  Int\n  open val numberOfItems: Int\n  fun clear()\n  fun initialize(newItem: dynamic): dynamic\n  fun\n  insertItemBefore(newItem: dynamic, index: Int): dynamic\n  fun replaceItem(newItem: dynamic, index: Int):\n  dynamic\n  fun removeItem(index: Int): dynamic\n  fun appendItem(newItem: dynamic): dynamic\n  fun

```

```

getItem(index: Int): dynamic\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGNameList.get(index: Int):
dynamic = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGNameList.set(index: Int,
newItem: dynamic) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript
[SVGNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGNumberList) to Kotlin\n */\npublic external
abstract class SVGNumberList {\n    open val length: Int\n    open val numberOfItems: Int\n    fun clear()\n    fun
initialize(newItem: SVGNumber): SVGNumber\n    fun insertItemBefore(newItem: SVGNumber, index: Int):
SVGNumber\n    fun replaceItem(newItem: SVGNumber, index: Int): SVGNumber\n    fun removeItem(index: Int):
SVGNumber\n    fun appendItem(newItem: SVGNumber): SVGNumber\n    fun getItem(index: Int):
SVGNumber\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGNumberList.get(index:
Int): SVGNumber? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGNumberList.set(index:
Int, newItem: SVGNumber) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript
[SVGLengthList](https://developer.mozilla.org/en/docs/Web/API/SVGLengthList) to Kotlin\n */\npublic external
abstract class SVGLengthList {\n    open val length: Int\n    open val numberOfItems: Int\n    fun clear()\n    fun
initialize(newItem: SVGLength): SVGLength\n    fun insertItemBefore(newItem: SVGLength, index: Int):
SVGLength\n    fun replaceItem(newItem: SVGLength, index: Int): SVGLength\n    fun removeItem(index: Int):
SVGLength\n    fun appendItem(newItem: SVGLength): SVGLength\n    fun getItem(index: Int):
SVGLength\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGLengthList.get(index:
Int): SVGLength? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGLengthList.set(index: Int,
newItem: SVGLength) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript
[SVGAnimatedBoolean](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedBoolean) to Kotlin\n */\npublic external abstract class SVGAnimatedBoolean {\n    open var baseVal: Boolean\n    open val animVal:
Boolean\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedEnumeration](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedEnumeration) to
Kotlin\n */\npublic external abstract class SVGAnimatedEnumeration {\n    open var baseVal: Short\n    open val
animVal: Short\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedInteger](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedInteger) to Kotlin\n */\npublic external abstract class SVGAnimatedInteger {\n    open var baseVal: Int\n    open val animVal:
Int\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedNumber](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedNumber) to Kotlin\n */\npublic external abstract class SVGAnimatedNumber {\n    open var baseVal: Float\n    open val animVal:
Float\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedLength](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedLength) to Kotlin\n */\npublic external abstract class SVGAnimatedLength {\n    open val baseVal: SVGLength\n    open val animVal:
SVGLength\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedAngle](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedAngle) to Kotlin\n */\npublic
external abstract class SVGAnimatedAngle {\n    open val baseVal: SVGAngle\n    open val animVal:
SVGAngle\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedString](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedString) to Kotlin\n */\npublic
external abstract class SVGAnimatedString {\n    open var baseVal: String\n    open val animVal: String\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedRect](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedRect) to Kotlin\n */\npublic external abstract class SVGAnimatedRect {\n    open val baseVal: DOMRect\n    open val
animVal: DOMRectReadOnly\n}\n\n/**\n * Exposes the JavaScript

```

[SVGAnimatedNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedNumberList) to Kotlin\n
`*\npublic external abstract class SVGAnimatedNumberList {\n open val baseVal: SVGNumberList\n open val animVal: SVGNumberList\n}\n\n/**\n * Exposes the JavaScript`

[SVGAnimatedLengthList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedLengthList) to Kotlin\n
`*\npublic external abstract class SVGAnimatedLengthList {\n open val baseVal: SVGLengthList\n open val animVal: SVGLengthList\n}\n\n/**\n * Exposes the JavaScript`

[SVGStringList](https://developer.mozilla.org/en/docs/Web/API/SVGStringList) to Kotlin\n
`*\npublic external abstract class SVGStringList {\n open val length: Int\n open val numberOfItems: Int\n fun clear()\n fun initialize(newItem: String): String\n fun insertItemBefore(newItem: String, index: Int): String\n fun replaceItem(newItem: String, index: Int): String\n fun removeItem(index: Int): String\n fun appendItem(newItem: String): String\n fun getItem(index: Int): String\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGStringList.get(index: Int): String? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGStringList.set(index: Int, newItem: String) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript`

[SVGUnitTypes](https://developer.mozilla.org/en/docs/Web/API/SVGUnitTypes) to Kotlin\n
`*\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface SVGUnitTypes {\n companion object {\n val SVG_UNIT_TYPE_UNKNOWN: Short\n val SVG_UNIT_TYPE_USERSPACEONUSE: Short\n val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short\n }\n}\n\n/**\n * Exposes the JavaScript`

[SVGTTests](https://developer.mozilla.org/en/docs/Web/API/SVGTTests) to Kotlin\n
`*\npublic external interface SVGTTests {\n val requiredExtensions: SVGStringList\n val systemLanguage: SVGStringList\n}\n\npublic external interface SVGFitToViewBox {\n val viewBox: SVGAnimatedRect\n val preserveAspectRatio: SVGAnimatedPreserveAspectRatio\n}\n\n/**\n * Exposes the JavaScript`

[SVGZoomAndPan](https://developer.mozilla.org/en/docs/Web/API/SVGZoomAndPan) to Kotlin\n
`*\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface SVGZoomAndPan {\n var zoomAndPan: Short\n\n companion object {\n val SVG_ZOOMANDPAN_UNKNOWN: Short\n val SVG_ZOOMANDPAN_DISABLE: Short\n val SVG_ZOOMANDPAN_MAGNIFY: Short\n }\n}\n\n/**\n * Exposes the JavaScript`

[SVGURIReference](https://developer.mozilla.org/en/docs/Web/API/SVGURIReference) to Kotlin\n
`*\npublic external interface SVGURIReference {\n val href: SVGAnimatedString\n}\n\n/**\n * Exposes the JavaScript`

[SVGSVGElement](https://developer.mozilla.org/en/docs/Web/API/SVGSVGElement) to Kotlin\n
`*\npublic external abstract class SVGSVGElement : SVGGraphicsElement, SVGFitToViewBox, SVGZoomAndPan, WindowEventHandlers {\n open val x: SVGAnimatedLength\n open val y: SVGAnimatedLength\n open val width: SVGAnimatedLength\n open val height: SVGAnimatedLength\n open var currentScale: Float\n open val currentTranslate: DOMPointReadOnly\n fun getIntersectionList(rect: DOMRectReadOnly, referenceElement: SVGElement?): NodeList\n fun getEnclosureList(rect: DOMRectReadOnly, referenceElement: SVGElement?): NodeList\n fun checkIntersection(element: SVGElement, rect: DOMRectReadOnly): Boolean\n fun checkEnclosure(element: SVGElement, rect: DOMRectReadOnly): Boolean\n fun deselectAll()\n fun createSVGNumber(): SVGNumber\n fun createSVGLength(): SVGLength\n fun createSVGAngle(): SVGAngle\n fun createSVGPoint(): DOMPoint\n fun createSVGMatrix(): DOMMatrix\n fun createSVGRect(): DOMRect\n fun createSVGTransform(): SVGTransform\n fun createSVGTransformFromMatrix(matrix: DOMMatrixReadOnly): SVGTransform\n fun getElementById(elementId: String): Element\n fun suspendRedraw(maxWaitMilliseconds: Int): Int\n fun unsuspendRedraw(suspendHandleID: Int)\n fun unsuspendRedrawAll()\n fun forceRedraw()\n\n companion object {\n val SVG_ZOOMANDPAN_UNKNOWN: Short\n val SVG_ZOOMANDPAN_DISABLE: Short\n val SVG_ZOOMANDPAN_MAGNIFY: Short\n val ELEMENT_NODE: Short\n val`

```

ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val
ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[SVGGElement](https://developer.mozilla.org/en/docs/Web/API/SVGGElement) to Kotlin \n * \n public external
abstract class SVGGElement : SVGGraphicsElement { \n    companion object { \n    val ELEMENT_NODE:
Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE:
Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n public external abstract class
SVGUnknownElement : SVGGraphicsElement { \n    companion object { \n    val ELEMENT_NODE: Short\n
val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n
val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[SVGDefsElement](https://developer.mozilla.org/en/docs/Web/API/SVGDefsElement) to Kotlin \n * \n public
external abstract class SVGDefsElement : SVGGraphicsElement { \n    companion object { \n    val
ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val
CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE:
Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[SVGDescElement](https://developer.mozilla.org/en/docs/Web/API/SVGDescElement) to Kotlin \n * \n public
external abstract class SVGDescElement : SVGElement { \n    companion object { \n    val ELEMENT_NODE:
Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE:
Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val

```


DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }n}\n\n/**\n * Exposes the JavaScript [SVGMetadataElement](https://developer.mozilla.org/en/docs/Web/API/SVGMetadataElement) to Kotlin\n */\npublic external abstract class SVGMetadataElement : SVGElement {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val

DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }n}\n\n/**\n * Exposes the JavaScript [SVGTitleElement](https://developer.mozilla.org/en/docs/Web/API/SVGTitleElement) to Kotlin\n */\npublic external abstract class SVGTitleElement : SVGElement {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val

DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }n}\n\n/**\n * Exposes the JavaScript [SVGSymbolElement](https://developer.mozilla.org/en/docs/Web/API/SVGSymbolElement) to Kotlin\n */\npublic external abstract class SVGSymbolElement : SVGGraphicsElement, SVGFitToViewBox {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val

DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }n}\n\n/**\n * Exposes the JavaScript [SVGUseElement](https://developer.mozilla.org/en/docs/Web/API/SVGUseElement) to Kotlin\n */\npublic external abstract class SVGUseElement : SVGGraphicsElement, SVGURIReference {\n open val x: SVGAnimatedLength\n open val y: SVGAnimatedLength\n open val width: SVGAnimatedLength\n open val height: SVGAnimatedLength\n open val instanceRoot: SVGElement?\n open val animatedInstanceRoot: SVGElement?\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val

DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n

```

    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external open class
SVGUseElementShadowRoot : ShadowRoot {\n    companion object {\n        val ELEMENT_NODE: Short\n
val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n
val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external interface
SVGElementInstance {\n    val correspondingElement: SVGElement?\n        get() = definedExternally\n    val
correspondingUseElement: SVGUseElement?\n        get() = definedExternally\n}\n\npublic external open class
ShadowAnimation(source: dynamic, newTarget: dynamic) {\n    open val sourceAnimation: dynamic\n}\n\n/**\n *
Exposes the JavaScript [SVGSwitchElement](https://developer.mozilla.org/en/docs/Web/API/SVGSwitchElement)
to Kotlin\n */\n\npublic external abstract class SVGSwitchElement : SVGGraphicsElement {\n    companion object
{\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n
val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val
ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE:
Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external interface
GetSVGDocument {\n    fun getSVGDocument(): Document\n}\n\n/**\n * Exposes the JavaScript
[SVGStyleElement](https://developer.mozilla.org/en/docs/Web/API/SVGStyleElement) to Kotlin\n */\n\npublic
external abstract class SVGStyleElement : SVGElement, LinkStyle {\n    open var type: String\n    open var media:
String\n    open var title: String\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGTransform](https://developer.mozilla.org/en/docs/Web/API/SVGTransform) to Kotlin\n */\n\npublic external
abstract class SVGTransform {\n    open val type: Short\n    open val matrix: DOMMatrix\n    open val angle:
Float\n    fun setMatrix(matrix: DOMMatrixReadOnly)\n    fun setTranslate(tx: Float, ty: Float)\n    fun setScale(sx:
Float, sy: Float)\n    fun setRotate(angle: Float, cx: Float, cy: Float)\n    fun setSkewX(angle: Float)\n    fun
setSkewY(angle: Float)\n\n    companion object {\n        val SVG_TRANSFORM_UNKNOWN: Short\n        val
SVG_TRANSFORM_MATRIX: Short\n        val SVG_TRANSFORM_TRANSLATE: Short\n        val
SVG_TRANSFORM_SCALE: Short\n        val SVG_TRANSFORM_ROTATE: Short\n        val
SVG_TRANSFORM_SKEWX: Short\n        val SVG_TRANSFORM_SKEWY: Short\n    }\n}\n\n/**\n * Exposes

```

the JavaScript [SVGTransformList](https://developer.mozilla.org/en/docs/Web/API/SVGTransformList) to Kotlin

```

*\npublic external abstract class SVGTransformList {\n  open val length: Int\n  open val numberOfItems: Int\n  fun clear()\n  fun initialize(newItem: SVGTransform): SVGTransform\n  fun insertItemBefore(newItem: SVGTransform, index: Int): SVGTransform\n  fun replaceItem(newItem: SVGTransform, index: Int): SVGTransform\n  fun removeItem(index: Int): SVGTransform\n  fun appendItem(newItem: SVGTransform): SVGTransform\n  fun createSVGTransformFromMatrix(matrix: DOMMatrixReadOnly): SVGTransform\n  fun consolidate(): SVGTransform?\n  fun getItem(index: Int): SVGTransform\n}\n\n@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGTransformList.get(index: Int): SVGTransform? = asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGTransformList.set(index: Int, newItem: SVGTransform) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript [SVGAnimatedTransformList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedTransformList) to Kotlin\n */\npublic external abstract class SVGAnimatedTransformList {\n  open val baseVal: SVGTransformList\n  open val animVal: SVGTransformList\n}\n\n/**\n * Exposes the JavaScript [SVGPreserveAspectRatio](https://developer.mozilla.org/en/docs/Web/API/SVGPreserveAspectRatio) to Kotlin\n */\npublic external abstract class SVGPreserveAspectRatio {\n  open var align: Short\n  open var meetOrSlice: Short\n}\n\ncompanion object {\n  val SVG_PRESERVEASPECTRATIO_UNKNOWN: Short\n  val SVG_PRESERVEASPECTRATIO_NONE: Short\n  val SVG_PRESERVEASPECTRATIO_XMINYMIN: Short\n  val SVG_PRESERVEASPECTRATIO_XMIDYMIN: Short\n  val SVG_PRESERVEASPECTRATIO_XMAXYMIN: Short\n  val SVG_PRESERVEASPECTRATIO_XMINYMID: Short\n  val SVG_PRESERVEASPECTRATIO_XMIDYMID: Short\n  val SVG_PRESERVEASPECTRATIO_XMAXYMID: Short\n  val SVG_PRESERVEASPECTRATIO_XMINYMAX: Short\n  val SVG_PRESERVEASPECTRATIO_XMIDYMAX: Short\n  val SVG_PRESERVEASPECTRATIO_XMAXYMAX: Short\n  val SVG_MEETORSLICE_UNKNOWN: Short\n  val SVG_MEETORSLICE_MEET: Short\n  val SVG_MEETORSLICE_SLICE: Short\n}\n\n/**\n * Exposes the JavaScript [SVGAnimatedPreserveAspectRatio](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedPreserveAspectRatio) to Kotlin\n */\npublic external abstract class SVGAnimatedPreserveAspectRatio {\n  open val baseVal: SVGPreserveAspectRatio\n  open val animVal: SVGPreserveAspectRatio\n}\n\n/**\n * Exposes the JavaScript [SVGPathElement](https://developer.mozilla.org/en/docs/Web/API/SVGPathElement) to Kotlin\n */\npublic external abstract class SVGPathElement : SVGGeometryElement {\n  companion object {\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n}\n\n/**\n * Exposes the JavaScript [SVGRectElement](https://developer.mozilla.org/en/docs/Web/API/SVGRectElement) to Kotlin\n */\npublic external abstract class SVGRectElement : SVGGeometryElement {\n  open val x: SVGAnimatedLength\n  open val y: SVGAnimatedLength\n  open val width: SVGAnimatedLength\n  open val height: SVGAnimatedLength\n  open val rx: SVGAnimatedLength\n  open val ry: SVGAnimatedLength\n}\n\ncompanion object {\n  val ELEMENT_NODE: Short\n  val ATTRIBUTE_NODE: Short\n  val TEXT_NODE: Short\n  val

```

```

CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE:
Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[SVGCircleElement](https://developer.mozilla.org/en/docs/Web/API/SVGCircleElement) to Kotlin \n * \n public
external abstract class SVGCircleElement : SVGGeometryElement { \n    open val cx: SVGAnimatedLength \n
open val cy: SVGAnimatedLength \n    open val r: SVGAnimatedLength \n \n    companion object { \n        val
ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[SVGEllipseElement](https://developer.mozilla.org/en/docs/Web/API/SVGEllipseElement) to Kotlin \n * \n public
external abstract class SVGEllipseElement : SVGGeometryElement { \n    open val cx: SVGAnimatedLength \n
open val cy: SVGAnimatedLength \n    open val rx: SVGAnimatedLength \n    open val ry: SVGAnimatedLength \n \n
    companion object { \n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val
TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE:
Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val
COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n
        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[SVGLineElement](https://developer.mozilla.org/en/docs/Web/API/SVGLineElement) to Kotlin \n * \n public
external abstract class SVGLineElement : SVGGeometryElement { \n    open val x1: SVGAnimatedLength \n    open
val y1: SVGAnimatedLength \n    open val x2: SVGAnimatedLength \n    open val y2: SVGAnimatedLength \n \n
    companion object { \n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val
TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE:
Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val
COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n
        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[SVGMeshElement](https://developer.mozilla.org/en/docs/Web/API/SVGMeshElement) to Kotlin \n * \n public
external abstract class SVGMeshElement : SVGGeometryElement, SVGURIReference { \n    companion object { \n
        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:

```

```

Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedPoints](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedPoints) to Kotlin\n
*/\npublic external interface SVGAnimatedPoints {\n    val points: SVGPointList\n    val animatedPoints:
SVGPointList\n}\n\npublic external abstract class SVGPointList {\n    open val length: Int\n    open val
numberOfItems: Int\n    fun clear()\n    fun initialize(newItem: DOMPoint): DOMPoint\n    fun
insertItemBefore(newItem: DOMPoint, index: Int): DOMPoint\n    fun replaceItem(newItem: DOMPoint, index:
Int): DOMPoint\n    fun removeItem(index: Int): DOMPoint\n    fun appendItem(newItem: DOMPoint):
DOMPoint\n    fun getItem(index: Int): DOMPoint\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGPointList.get(index: Int):
DOMPoint? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGPointList.set(index: Int,
newItem: DOMPoint) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript
[SVGPolylineElement](https://developer.mozilla.org/en/docs/Web/API/SVGPolylineElement) to Kotlin\n
*/\npublic external abstract class SVGPolylineElement : SVGGeometryElement, SVGAnimatedPoints {\n
companion object {\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val
TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE:
Short\n    val ENTITY_NODE: Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val
COMMENT_NODE: Short\n    val DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n
    val DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGPolygonElement](https://developer.mozilla.org/en/docs/Web/API/SVGPolygonElement) to Kotlin\n
*/\npublic external abstract class SVGPolygonElement : SVGGeometryElement, SVGAnimatedPoints {\n
companion object {\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val
TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE:
Short\n    val ENTITY_NODE: Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val
COMMENT_NODE: Short\n    val DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n
    val DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGTextContentElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextContentElement) to Kotlin\n
*/\npublic external abstract class SVGTextContentElement : SVGGraphicsElement {\n    open val textLength:
SVGAnimatedLength\n    open val lengthAdjust: SVGAnimatedEnumeration\n    fun getNumberOfChars(): Int\n
fun getComputedTextLength(): Float\n    fun getSubStringLength(charnum: Int, nchars: Int): Float\n    fun
getStartPositionOfChar(charnum: Int): DOMPoint\n    fun getEndPositionOfChar(charnum: Int): DOMPoint\n    fun
getExtentOfChar(charnum: Int): DOMRect\n    fun getRotationOfChar(charnum: Int): Float\n    fun
getCharNumAtPosition(point: DOMPoint): Int\n    fun selectSubString(charnum: Int, nchars: Int)\n}\n\ncompanion
object {\n    val LENGTHADJUST_UNKNOWN: Short\n    val LENGTHADJUST_SPACING: Short\n

```

```

val LENGTHADJUST_SPACINGANDGLYPHS: Short\n    val ELEMENT_NODE: Short\n    val
ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val
ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGTextPositioningElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextPositioningElement) to
Kotlin\n */\npublic external abstract class SVGTextPositioningElement : SVGTextContentElement {\n    open val x:
SVGAnimatedLengthList\n    open val y: SVGAnimatedLengthList\n    open val dx: SVGAnimatedLengthList\n
open val dy: SVGAnimatedLengthList\n    open val rotate: SVGAnimatedNumberList\n\n    companion object {\n
    val LENGTHADJUST_UNKNOWN: Short\n    val LENGTHADJUST_SPACING: Short\n    val
LENGTHADJUST_SPACINGANDGLYPHS: Short\n    val ELEMENT_NODE: Short\n    val
ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val
ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGTextElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextElement) to Kotlin\n */\npublic
external abstract class SVGTextElement : SVGTextPositioningElement {\n    companion object {\n    val
LENGTHADJUST_UNKNOWN: Short\n    val LENGTHADJUST_SPACING: Short\n    val
LENGTHADJUST_SPACINGANDGLYPHS: Short\n    val ELEMENT_NODE: Short\n    val
ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val
ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGTSpanElement](https://developer.mozilla.org/en/docs/Web/API/SVGTSpanElement) to Kotlin\n */\npublic
external abstract class SVGTSpanElement : SVGTextPositioningElement {\n    companion object {\n    val
LENGTHADJUST_UNKNOWN: Short\n    val LENGTHADJUST_SPACING: Short\n    val
LENGTHADJUST_SPACINGANDGLYPHS: Short\n    val ELEMENT_NODE: Short\n    val
ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val
ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n

```

```

    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGTextPathElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextPathElement) to Kotlin\n
*/\npublic external abstract class SVGTextPathElement : SVGTextContentElement, SVGURIReference {\n    open
val startOffset: SVGAnimatedLength\n    open val method: SVGAnimatedEnumeration\n    open val spacing:
SVGAnimatedEnumeration\n\n    companion object {\n        val TEXTPATH_METHODTYPE_UNKNOWN:
Short\n        val TEXTPATH_METHODTYPE_ALIGN: Short\n        val
TEXTPATH_METHODTYPE_STRETCH: Short\n        val TEXTPATH_SPACINGTYPE_UNKNOWN: Short\n
        val TEXTPATH_SPACINGTYPE_AUTO: Short\n        val TEXTPATH_SPACINGTYPE_EXACT: Short\n
        val LENGTHADJUST_UNKNOWN: Short\n        val LENGTHADJUST_SPACING: Short\n        val
LENGTHADJUST_SPACINGANDGLYPHS: Short\n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGImageElement](https://developer.mozilla.org/en/docs/Web/API/SVGImageElement) to Kotlin\n
*/\npublic external abstract class SVGImageElement : SVGGraphicsElement, SVGURIReference,
HTMLOrSVGImageElement {\n    open val x: SVGAnimatedLength\n    open val y: SVGAnimatedLength\n    open
val width: SVGAnimatedLength\n    open val height: SVGAnimatedLength\n    open val preserveAspectRatio:
SVGAnimatedPreserveAspectRatio\n    open var crossOrigin: String?\n\n    companion object {\n        val
ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGForeignObjectElement](https://developer.mozilla.org/en/docs/Web/API/SVGForeignObjectElement) to
Kotlin\n
*/\npublic external abstract class SVGForeignObjectElement : SVGGraphicsElement {\n    open val x:
SVGAnimatedLength\n    open val y: SVGAnimatedLength\n    open val width: SVGAnimatedLength\n    open val
height: SVGAnimatedLength\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external abstract class

```



```

DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external abstract class
SVGMeshpatchElement : SVGElement {\n    companion object {\n        val ELEMENT_NODE: Short\n    val
ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val
ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGStopElement](https://developer.mozilla.org/en/docs/Web/API/SVGStopElement) to Kotlin\n */\n\npublic
external abstract class SVGStopElement : SVGElement {\n    open val offset: SVGAnimatedNumber\n\n
companion object {\n        val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val
TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE:
Short\n    val ENTITY_NODE: Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val
COMMENT_NODE: Short\n    val DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n
    val DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGPatternElement](https://developer.mozilla.org/en/docs/Web/API/SVGPatternElement) to Kotlin\n */\n\npublic
external abstract class SVGPatternElement : SVGElement, SVGFitToViewBox, SVGURIReference,
SVGUnitTypes {\n    open val patternUnits: SVGAnimatedEnumeration\n    open val patternContentUnits:
SVGAnimatedEnumeration\n    open val patternTransform: SVGAnimatedTransformList\n    open val x:
SVGAnimatedLength\n    open val y: SVGAnimatedLength\n    open val width: SVGAnimatedLength\n    open val
height: SVGAnimatedLength\n\n    companion object {\n        val SVG_UNIT_TYPE_UNKNOWN: Short\n
    val SVG_UNIT_TYPE_USERSPACEONUSE: Short\n    val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX:
Short\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n
    val CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val
ENTITY_NODE: Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE:
Short\n    val DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external abstract class
SVGHatchElement : SVGElement {\n    companion object {\n        val ELEMENT_NODE: Short\n    val
ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val
ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val

```



```

DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }n}\n\n/**\n * Exposes the JavaScript
[SVGViewElement](https://developer.mozilla.org/en/docs/Web/API/SVGViewElement) to Kotlin\n *\npublic
external abstract class SVGViewElement : SVGElement, SVGFitToViewBox, SVGZoomAndPan {\n    companion
object {\n        val SVG_ZOOMANDPAN_UNKNOWN: Short\n        val SVG_ZOOMANDPAN_DISABLE:
Short\n        val SVG_ZOOMANDPAN_MAGNIFY: Short\n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }n}\n\n"/\n * Copyright 2010-2021
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-
GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.files\n\nimport
kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\nimport
org.w3c.xhr.*\n\n/**\n * Exposes the JavaScript [Blob](https://developer.mozilla.org/en/docs/Web/API/Blob) to
Kotlin\n *\npublic external open class Blob(blobParts: Array<dynamic> = definedExternally, options:
BlobPropertyBag = definedExternally) : MediaProvider, ImageBitmapSource {\n    open val size: Number\n    open
val type: String\n    open val isClosed: Boolean\n    fun slice(start: Int = definedExternally, end: Int =
definedExternally, contentType: String = definedExternally): Blob\n    fun close()\n}\n\npublic external interface
BlobPropertyBag {\n    var type: String? /* = "" */\n        get() = definedExternally\n        set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun BlobPropertyBag(type: String? = ""):
BlobPropertyBag {\n    val o = js("{}")\n    o["type"] = type\n    return o\n}\n\n/**\n * Exposes the JavaScript
[File](https://developer.mozilla.org/en/docs/Web/API/File) to Kotlin\n *\npublic external open class File(fileBits:
Array<dynamic>, fileName: String, options: FilePropertyBag = definedExternally) : Blob {\n    open val name:
String\n    open val lastModified: Int\n}\n\npublic external interface FilePropertyBag : BlobPropertyBag {\n    var
lastModified: Int?\n        get() = definedExternally\n        set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun FilePropertyBag(lastModified: Int? =
undefined, type: String? = ""): FilePropertyBag {\n    val o = js("{}")\n    o["lastModified"] = lastModified\n
o["type"] = type\n    return o\n}\n\n/**\n * Exposes the JavaScript
[FileList](https://developer.mozilla.org/en/docs/Web/API/FileList) to Kotlin\n *\npublic external abstract class
FileList : ItemArrayLike<File> {\n    override fun item(index: Int):
File?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun FileList.get(index: Int): File?
= asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[FileReader](https://developer.mozilla.org/en/docs/Web/API/FileReader) to Kotlin\n *\npublic external open class
FileReader : EventTarget {\n    open val readyState: Short\n    open val result: dynamic\n    open val error:
dynamic\n    var onloadstart: ((ProgressEvent) -> dynamic)?\n    var onprogress: ((ProgressEvent) -> dynamic)?\n
var onload: ((Event) -> dynamic)?\n    var onabort: ((Event) -> dynamic)?\n    var onerror: ((Event) ->
dynamic)?\n

```



```

o["renotify"] = notify\n o["silent"] = silent\n o["noscreen"] = noscreen\n o["requireInteraction"] =
requireInteraction\n o["sticky"] = sticky\n o["data"] = data\n o["actions"] = actions\n return
o\n}\n\npublic external interface NotificationAction {\n var action: String?\n var title: String?\n var icon:
String?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun NotificationAction(action: String?,
title: String?, icon: String? = undefined): NotificationAction {\n val o = js("{}")\n o["action"] = action\n
o["title"] = title\n o["icon"] = icon\n return o\n}\n\npublic external interface GetNotificationOptions {\n var
tag: String? /* = "" */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun GetNotificationOptions(tag: String? =
""): GetNotificationOptions {\n val o = js("{}")\n o["tag"] = tag\n return o\n}\n\n/**\n * Exposes the
JavaScript [NotificationEvent](https://developer.mozilla.org/en/docs/Web/API/NotificationEvent) to Kotlin\n
*/\n\npublic external open class NotificationEvent(type: String, eventInitDict: NotificationEventInit) :
ExtendableEvent {\n open val notification: Notification\n open val action: String\n\n companion object {\n
val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val
BUBBLING_PHASE: Short\n }\n}\n\npublic external interface NotificationEventInit : ExtendableEventInit {\n
var notification: Notification?\n var action: String? /* = "" */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun NotificationEventInit(notification:
Notification?, action: String? = "", bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? =
false): NotificationEventInit {\n val o = js("{}")\n o["notification"] = notification\n o["action"] =
action\n o["bubbles"] = bubbles\n o["cancelable"] = cancelable\n o["composed"] = composed\n return
o\n}\n\n/* please, don't implement this interface!
*/\n\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external
interface NotificationPermission {\n companion object\n}\n\npublic inline val
NotificationPermission.Companion.DEFAULT: NotificationPermission get() =
"default".asDynamic().unsafeCast<NotificationPermission>()\n\npublic inline val
NotificationPermission.Companion.DENIED: NotificationPermission get() =
"denied".asDynamic().unsafeCast<NotificationPermission>()\n\npublic inline val
NotificationPermission.Companion.GRANTED: NotificationPermission get() =
"granted".asDynamic().unsafeCast<NotificationPermission>()\n\n/* please, don't implement this interface!
*/\n\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external
interface NotificationDirection {\n companion object\n}\n\npublic inline val
NotificationDirection.Companion.AUTO: NotificationDirection get() =
"auto".asDynamic().unsafeCast<NotificationDirection>()\n\npublic inline val
NotificationDirection.Companion.LTR: NotificationDirection get() =
"ltr".asDynamic().unsafeCast<NotificationDirection>()\n\npublic inline val
NotificationDirection.Companion.RTL: NotificationDirection get() =
"rtl".asDynamic().unsafeCast<NotificationDirection>()"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n\n//
See github.com/kotlin/dukat for details\n\npackage org.w3c.workers\n\nimport kotlin.js.*\nimport
org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\nimport org.w3c.fetch.*\nimport
org.w3c.notifications.*\n\n/**\n * Exposes the JavaScript
[ServiceWorker](https://developer.mozilla.org/en/docs/Web/API/ServiceWorker) to Kotlin\n */\n\npublic external
abstract class ServiceWorker : EventTarget, AbstractWorker, UnionMessagePortOrServiceWorker,
UnionClientOrMessagePortOrServiceWorker {\n open val scriptURL: String\n open val state:

```

```

ServiceWorkerState\n  open var onstatechange: ((Event) -> dynamic)?\n  fun postMessage(message: Any?,
transfer: Array<dynamic> = definedExternally)\n}\n\n/**\n * Exposes the JavaScript
[ServiceWorkerRegistration](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerRegistration) to
Kotlin\n */\npublic external abstract class ServiceWorkerRegistration : EventTarget {\n  open val installing:
ServiceWorker?\n  open val waiting: ServiceWorker?\n  open val active: ServiceWorker?\n  open val scope:
String\n  open var onupdatefound: ((Event) -> dynamic)?\n  open val APISpace: dynamic\n  fun update():
Promise<Unit>\n  fun unregister(): Promise<Boolean>\n  fun showNotification(title: String, options:
NotificationOptions = definedExternally): Promise<Unit>\n  fun getNotifications(filter: GetNotificationOptions =
definedExternally): Promise<Array<Notification>>\n  fun methodName(): Promise<dynamic>\n}\n\n/**\n *
Exposes the JavaScript
[ServiceWorkerContainer](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerContainer) to Kotlin\n
*/\npublic external abstract class ServiceWorkerContainer : EventTarget {\n  open val controller:
ServiceWorker?\n  open val ready: Promise<ServiceWorkerRegistration>\n  open var oncontrollerchange:
((Event) -> dynamic)?\n  open var onmessage: ((MessageEvent) -> dynamic)?\n  fun register(scriptURL: String,
options: RegistrationOptions = definedExternally): Promise<ServiceWorkerRegistration>\n  fun
getRegistration(clientURL: String = definedExternally): Promise<Any?>\n  fun getRegistrations():
Promise<Array<ServiceWorkerRegistration>>\n  fun startMessages()\n}\n\npublic external interface
RegistrationOptions {\n  var scope: String?\n  get() = definedExternally\n  set(value) = definedExternally\n
var type: WorkerType? /* = WorkerType.CLASSIC */\n  get() = definedExternally\n  set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun RegistrationOptions(scope: String? =
undefined, type: WorkerType? = WorkerType.CLASSIC): RegistrationOptions {\n  val o = js(\"({})\")\n
o[\"scope\"] = scope\n o[\"type\"] = type\n return o\n}\n\n/**\n * Exposes the JavaScript
[ServiceWorkerMessageEvent](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerMessageEvent) to
Kotlin\n */\npublic external open class ServiceWorkerMessageEvent(type: String, eventInitDict:
ServiceWorkerMessageEventInit = definedExternally) : Event {\n  open val data: Any?\n  open val origin:
String\n  open val lastEventId: String\n  open val source: UnionMessagePortOrServiceWorker?\n  open val
ports: Array<out MessagePort>?\n\n  companion object {\n    val NONE: Short\n    val
CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val BUBBLING_PHASE: Short\n
}\n}\n\npublic external interface ServiceWorkerMessageEventInit : EventInit {\n  var data: Any?\n  get() =
definedExternally\n  set(value) = definedExternally\n  var origin: String?\n  get() = definedExternally\n
set(value) = definedExternally\n  var lastEventId: String?\n  get() = definedExternally\n  set(value) =
definedExternally\n  var source: UnionMessagePortOrServiceWorker?\n  get() = definedExternally\n
set(value) = definedExternally\n  var ports: Array<MessagePort>?\n  get() = definedExternally\n  set(value)
= definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ServiceWorkerMessageEventInit(data:
Any? = undefined, origin: String? = undefined, lastEventId: String? = undefined, source:
UnionMessagePortOrServiceWorker? = undefined, ports: Array<MessagePort>? = undefined, bubbles: Boolean? =
false, cancelable: Boolean? = false, composed: Boolean? = false): ServiceWorkerMessageEventInit {\n  val o =
js(\"({})\")\n  o[\"data\"] = data\n  o[\"origin\"] = origin\n  o[\"lastEventId\"] = lastEventId\n  o[\"source\"] =
source\n  o[\"ports\"] = ports\n  o[\"bubbles\"] = bubbles\n  o[\"cancelable\"] = cancelable\n  o[\"composed\"] =
composed\n  return o\n}\n\n/**\n * Exposes the JavaScript
[ServiceWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerGlobalScope) to
Kotlin\n */\npublic external abstract class ServiceWorkerGlobalScope : WorkerGlobalScope {\n  open val clients:
Clients\n  open val registration: ServiceWorkerRegistration\n  open var oninstall: ((Event) -> dynamic)?\n  open
var onactivate: ((Event) -> dynamic)?\n  open var onfetch: ((FetchEvent) -> dynamic)?\n  open var
onforeignfetch: ((Event) -> dynamic)?\n  open var onmessage: ((MessageEvent) -> dynamic)?\n  open var
onnotificationclick: ((NotificationEvent) -> dynamic)?\n  open var onnotificationclose: ((NotificationEvent) ->

```

```

dynamic)?\n  open var onfunctionalevent: ((Event) -> dynamic)?\n  fun skipWaiting():
Promise<Unit>\n}\n\n/**\n * Exposes the JavaScript
[Client](https://developer.mozilla.org/en/docs/Web/API/Client) to Kotlin\n *\npublic external abstract class Client :
UnionClientOrMessagePortOrServiceWorker {\n  open val url: String\n  open val frameType: FrameType\n
open val id: String\n  fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)\n}\n\n/**\n
* Exposes the JavaScript [WindowClient](https://developer.mozilla.org/en/docs/Web/API/WindowClient) to
Kotlin\n *\npublic external abstract class WindowClient : Client {\n  open val visibilityState: dynamic\n  open
val focused: Boolean\n  fun focus(): Promise<WindowClient>\n  fun navigate(url: String):
Promise<WindowClient>\n}\n\n/**\n * Exposes the JavaScript
[Clients](https://developer.mozilla.org/en/docs/Web/API/Clients) to Kotlin\n *\npublic external abstract class
Clients {\n  fun get(id: String): Promise<Any?>\n  fun matchAll(options: ClientQueryOptions =
definedExternally): Promise<Array<Client>>\n  fun openWindow(url: String): Promise<WindowClient?>\n  fun
claim(): Promise<Unit>\n}\n\npublic external interface ClientQueryOptions {\n  var includeUncontrolled:
Boolean? /* = false */\n  get() = definedExternally\n  set(value) = definedExternally\n  var type:
ClientType? /* = ClientType.WINDOW */\n  get() = definedExternally\n  set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\npublic inline fun
ClientQueryOptions(includeUncontrolled: Boolean? = false, type: ClientType? = ClientType.WINDOW):
ClientQueryOptions {\n  val o = js(\"({})\")\n  o[\"includeUncontrolled\"] = includeUncontrolled\n  o[\"type\"] =
type\n  return o\n}\n\n/**\n * Exposes the JavaScript
[ExtendableEvent](https://developer.mozilla.org/en/docs/Web/API/ExtendableEvent) to Kotlin\n *\npublic external
open class ExtendableEvent(type: String, eventInitDict: ExtendableEventInit = definedExternally) : Event {\n  fun
waitUntil(f: Promise<Any?>)\n\n  companion object {\n    val NONE: Short\n    val CAPTURING_PHASE:
Short\n    val AT_TARGET: Short\n    val BUBBLING_PHASE: Short\n  }\n}\n\npublic external interface
ExtendableEventInit : EventInit\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\npublic inline fun ExtendableEventInit(bubbles:
Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): ExtendableEventInit {\n  val o =
js(\"({})\")\n  o[\"bubbles\"] = bubbles\n  o[\"cancelable\"] = cancelable\n  o[\"composed\"] = composed\n
return o\n}\n\n/**\n * Exposes the JavaScript
[InstallEvent](https://developer.mozilla.org/en/docs/Web/API/InstallEvent) to Kotlin\n *\npublic external open
class InstallEvent(type: String, eventInitDict: ExtendableEventInit = definedExternally) : ExtendableEvent {\n  fun
registerForeignFetch(options: ForeignFetchOptions)\n\n  companion object {\n    val NONE: Short\n    val
CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val BUBBLING_PHASE: Short\n  }\n}\n\npublic external interface ForeignFetchOptions {\n  var scopes: Array<String>?\n  var origins:
Array<String>?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\npublic inline fun ForeignFetchOptions(scopes:
Array<String>?, origins: Array<String>?): ForeignFetchOptions {\n  val o = js(\"({})\")\n  o[\"scopes\"] =
scopes\n  o[\"origins\"] = origins\n  return o\n}\n\n/**\n * Exposes the JavaScript
[FetchEvent](https://developer.mozilla.org/en/docs/Web/API/FetchEvent) to Kotlin\n *\npublic external open class
FetchEvent(type: String, eventInitDict: FetchEventInit) : ExtendableEvent {\n  open val request: Request\n  open
val clientId: String?\n  open val isReload: Boolean\n  fun respondWith(r: Promise<Response>)\n\n  companion
object {\n    val NONE: Short\n    val CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val
BUBBLING_PHASE: Short\n  }\n}\n\npublic external interface FetchEventInit : ExtendableEventInit {\n  var
request: Request?\n  var clientId: String? /* = null */\n  get() = definedExternally\n  set(value) =
definedExternally\n  var isReload: Boolean? /* = false */\n  get() = definedExternally\n  set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\npublic inline fun FetchEventInit(request: Request?,
clientId: String? = null, isReload: Boolean? = false, bubbles: Boolean? = false, cancelable: Boolean? = false,

```



```

composed: Boolean? = false): FetchEventInit {\n  val o = js("{}")\n  o["request"] = request\n  o["clientId"]
= clientId\n  o["isReload"] = isReload\n  o["bubbles"] = bubbles\n  o["cancelable"] = cancelable\n
o["composed"] = composed\n  return o\n}\n\npublic external open class ForeignFetchEvent(type: String,
eventInitDict: ForeignFetchEventInit) : ExtendableEvent {\n  open val request: Request\n  open val origin:
String\n  fun respondWith(r: Promise<ForeignFetchResponse>)\n\n  companion object {\n    val NONE:
Short\n    val CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val BUBBLING_PHASE:
Short\n  }\n}\n\npublic external interface ForeignFetchEventInit : ExtendableEventInit {\n  var request:
Request?\n  var origin: String? /* = "null" */\n  get() = definedExternally\n  set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun ForeignFetchEventInit(request:
Request?, origin: String? = "null", bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? =
false): ForeignFetchEventInit {\n  val o = js("{}")\n  o["request"] = request\n  o["origin"] = origin\n
o["bubbles"] = bubbles\n  o["cancelable"] = cancelable\n  o["composed"] = composed\n  return
o\n}\n\npublic external interface ForeignFetchResponse {\n  var response: Response?\n  var origin: String?\n
get() = definedExternally\n  set(value) = definedExternally\n  var headers: Array<String>?\n  get() =
definedExternally\n  set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun ForeignFetchResponse(response:
Response?, origin: String? = undefined, headers: Array<String>? = undefined): ForeignFetchResponse {\n  val o =
js("{}")\n  o["response"] = response\n  o["origin"] = origin\n  o["headers"] = headers\n  return
o\n}\n\n/**\n * Exposes the JavaScript
[ExtendableMessageEvent](https://developer.mozilla.org/en/docs/Web/API/ExtendableMessageEvent) to Kotlin\n
*/\n\npublic external open class ExtendableMessageEvent(type: String, eventInitDict: ExtendableMessageEventInit =
definedExternally) : ExtendableEvent {\n  open val data: Any?\n  open val origin: String\n  open val lastEventId:
String\n  open val source: UnionClientOrMessagePortOrServiceWorker?\n  open val ports: Array<out
MessagePort>?\n\n  companion object {\n    val NONE: Short\n    val CAPTURING_PHASE: Short\n    val
AT_TARGET: Short\n    val BUBBLING_PHASE: Short\n  }\n}\n\npublic external interface
ExtendableMessageEventInit : ExtendableEventInit {\n  var data: Any?\n  get() = definedExternally\n
set(value) = definedExternally\n  var origin: String?\n  get() = definedExternally\n  set(value) =
definedExternally\n  var lastEventId: String?\n  get() = definedExternally\n  set(value) =
definedExternally\n  var source: UnionClientOrMessagePortOrServiceWorker?\n  get() = definedExternally\n
set(value) = definedExternally\n  var ports: Array<MessagePort>?\n  get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun ExtendableMessageEventInit(data:
Any? = undefined, origin: String? = undefined, lastEventId: String? = undefined, source:
UnionClientOrMessagePortOrServiceWorker? = undefined, ports: Array<MessagePort>? = undefined, bubbles:
Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): ExtendableMessageEventInit {\n
val o = js("{}")\n  o["data"] = data\n  o["origin"] = origin\n  o["lastEventId"] = lastEventId\n
o["source"] = source\n  o["ports"] = ports\n  o["bubbles"] = bubbles\n  o["cancelable"] = cancelable\n
o["composed"] = composed\n  return o\n}\n\n/**\n * Exposes the JavaScript
[Cache](https://developer.mozilla.org/en/docs/Web/API/Cache) to Kotlin\n
*/\n\npublic external abstract class Cache
{\n  fun match(request: dynamic, options: CacheQueryOptions = definedExternally): Promise<Any?>\n  fun
matchAll(request: dynamic = definedExternally, options: CacheQueryOptions = definedExternally):
Promise<Array<Response>>\n  fun add(request: dynamic): Promise<Unit>\n  fun addAll(requests:
Array<dynamic>): Promise<Unit>\n  fun put(request: dynamic, response: Response): Promise<Unit>\n  fun
delete(request: dynamic, options: CacheQueryOptions = definedExternally): Promise<Boolean>\n  fun
keys(request: dynamic = definedExternally, options: CacheQueryOptions = definedExternally):
Promise<Array<Request>>\n}\n\npublic external interface CacheQueryOptions {\n  var ignoreSearch: Boolean? /*
= false */\n  get() = definedExternally\n  set(value) = definedExternally\n  var ignoreMethod: Boolean? /*
=

```

```

false *^n    get() = definedExternally\n    set(value) = definedExternally\n    var ignoreVary: Boolean? /* =
false *^n    get() = definedExternally\n    set(value) = definedExternally\n    var cacheName: String?\n
get() = definedExternally\n    set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun CacheQueryOptions(ignoreSearch:
Boolean? = false, ignoreMethod: Boolean? = false, ignoreVary: Boolean? = false, cacheName: String? = undefined):
CacheQueryOptions {\n    val o = js(\"({})\")\n    o[\"ignoreSearch\"] = ignoreSearch\n    o[\"ignoreMethod\"] =
ignoreMethod\n    o[\"ignoreVary\"] = ignoreVary\n    o[\"cacheName\"] = cacheName\n    return o\n}\n\npublic
external interface CacheBatchOperation {\n    var type: String?\n    get() = definedExternally\n    set(value) =
definedExternally\n    var request: Request?\n    get() = definedExternally\n    set(value) = definedExternally\n
var response: Response?\n    get() = definedExternally\n    set(value) = definedExternally\n    var options:
CacheQueryOptions?\n    get() = definedExternally\n    set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun CacheBatchOperation(type: String? =
undefined, request: Request? = undefined, response: Response? = undefined, options: CacheQueryOptions? =
undefined): CacheBatchOperation {\n    val o = js(\"({})\")\n    o[\"type\"] = type\n    o[\"request\"] = request\n
o[\"response\"] = response\n    o[\"options\"] = options\n    return o\n}\n\n/**\n * Exposes the JavaScript
[CacheStorage](https://developer.mozilla.org/en/docs/Web/API/CacheStorage) to Kotlin\n */\n\npublic external
abstract class CacheStorage {\n    fun match(request: dynamic, options: CacheQueryOptions = definedExternally):
Promise<Any?>\n    fun has(cacheName: String): Promise<Boolean>\n    fun open(cacheName: String):
Promise<Cache>\n    fun delete(cacheName: String): Promise<Boolean>\n    fun keys():
Promise<Array<String>>\n}\n\npublic external open class FunctionalEvent : ExtendableEvent {\n    companion
object {\n        val NONE: Short\n        val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val
BUBBLING_PHASE: Short\n    }\n}\n\npublic external interface UnionMessagePortOrServiceWorker\n\npublic
external interface UnionClientOrMessagePortOrServiceWorker\n\n/* please, don't implement this interface!
*/\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface ServiceWorkerState {\n    companion object\n}\n\npublic inline val
ServiceWorkerState.Companion.INSTALLING: ServiceWorkerState get() =
\"installing\".asDynamic().unsafeCast<ServiceWorkerState>()\n\npublic inline val
ServiceWorkerState.Companion.INSTALLED: ServiceWorkerState get() =
\"installed\".asDynamic().unsafeCast<ServiceWorkerState>()\n\npublic inline val
ServiceWorkerState.Companion.ACTIVATING: ServiceWorkerState get() =
\"activating\".asDynamic().unsafeCast<ServiceWorkerState>()\n\npublic inline val
ServiceWorkerState.Companion.ACTIVATED: ServiceWorkerState get() =
\"activated\".asDynamic().unsafeCast<ServiceWorkerState>()\n\npublic inline val
ServiceWorkerState.Companion.REDUNDANT: ServiceWorkerState get() =
\"redundant\".asDynamic().unsafeCast<ServiceWorkerState>()\n\n/* please, don't implement this interface!
*/\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface FrameType {\n    companion object\n}\n\npublic inline val FrameType.Companion.AUXILIARY:
FrameType get() = \"auxiliary\".asDynamic().unsafeCast<FrameType>()\n\npublic inline val
FrameType.Companion.TOP_LEVEL: FrameType get() = \"top-
level\".asDynamic().unsafeCast<FrameType>()\n\npublic inline val FrameType.Companion.NESTED: FrameType
get() = \"nested\".asDynamic().unsafeCast<FrameType>()\n\npublic inline val FrameType.Companion.NONE:
FrameType get() = \"none\".asDynamic().unsafeCast<FrameType>()\n\n/* please, don't implement this interface!
*/\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface ClientType {\n    companion object\n}\n\npublic inline val ClientType.Companion.WINDOW: ClientType
get() = \"window\".asDynamic().unsafeCast<ClientType>()\n\npublic inline val ClientType.Companion.WORKER:
ClientType get() = \"worker\".asDynamic().unsafeCast<ClientType>()\n\npublic inline val
ClientType.Companion.SHAREDWORKER: ClientType get() =

```



```

"\.asDynamic().unsafeCast<XMLHttpRequestResponseType>()\n\npublic inline val
XMLHttpRequestResponseType.Companion.ARRAYBUFFER: XMLHttpRequestResponseType get() =
"arraybuffer".asDynamic().unsafeCast<XMLHttpRequestResponseType>()\n\npublic inline val
XMLHttpRequestResponseType.Companion.BLOB: XMLHttpRequestResponseType get() =
"blob".asDynamic().unsafeCast<XMLHttpRequestResponseType>()\n\npublic inline val
XMLHttpRequestResponseType.Companion.DOCUMENT: XMLHttpRequestResponseType get() =
"document".asDynamic().unsafeCast<XMLHttpRequestResponseType>()\n\npublic inline val
XMLHttpRequestResponseType.Companion.JSON: XMLHttpRequestResponseType get() =
"json".asDynamic().unsafeCast<XMLHttpRequestResponseType>()\n\npublic inline val
XMLHttpRequestResponseType.Companion.TEXT: XMLHttpRequestResponseType get() =
"text".asDynamic().unsafeCast<XMLHttpRequestResponseType>()","/>\n * Copyright 2010-2018 JetBrains s.r.o.
and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\nimport
kotlin.annotation.AnnotationRetention.BINARY\n\nimport kotlin.annotation.AnnotationRetention.SOURCE\n\nimport
kotlin.annotation.AnnotationTarget.*\n\nimport kotlin.internal.RequireKotlin\n\nimport
kotlin.internal.RequireKotlinVersionKind\n\nimport kotlin.reflect.KClass\n\n\n/**\n * Signals that the annotated
annotation class is a marker of an experimental API.\n *\n * Any declaration annotated with that marker is
considered an experimental declaration\n * and its call sites should accept the experimental aspect of it either by
using [UseExperimental],\n * or by being annotated with that marker themselves, effectively causing further
propagation of that experimental aspect.\n *\n * This class is deprecated in favor of a more general approach
provided by [RequiresOptIn]/[OptIn].\n
*/\n\n@Target(ANNOTATION_CLASS)\n@Retention(BINARY)\n@SinceKotlin("1.2")\n@DeprecatedSinceKotli
n(warningSince = "1.4", errorSince = "1.6")\n@Deprecated("Please use RequiresOptIn instead.")\npublic
annotation class Experimental(val level: Level = Level.ERROR) {\n    /**\n     * Severity of the diagnostic that
should be reported on usages of experimental API which did not explicitly accept the experimental aspect\n     * of
that API either by using [UseExperimental] or by being annotated with the corresponding marker annotation.\n
*/\n    public enum class Level {\n        /** Specifies that a warning should be reported on incorrect usages of this
experimental API. */\n        WARNING,\n        /** Specifies that an error should be reported on incorrect usages of
this experimental API. */\n        ERROR,\n    }\n}\n\n/**\n * Allows to use experimental API denoted by the given
markers in the annotated file, declaration, or expression.\n * If a declaration is annotated with [UseExperimental],
its usages are **not** required to opt-in to that experimental API.\n *\n * This class is deprecated in favor of a more
general approach provided by [RequiresOptIn]/[OptIn].\n */\n\n@Target(\n    CLASS, PROPERTY,
LOCAL_VARIABLE, VALUE_PARAMETER, CONSTRUCTOR, FUNCTION, PROPERTY_GETTER,
PROPERTY_SETTER, EXPRESSION, FILE,
TYPEALIAS)\n\n@Retention(SOURCE)\n@SinceKotlin("1.2")\n@DeprecatedSinceKotlin(warningSince =
"1.4", errorSince = "1.6")\n@Deprecated("Please use OptIn instead.", ReplaceWith("OptIn(*markerClass)",
"kotlin.OptIn"))\npublic annotation class UseExperimental(\n    vararg val markerClass: KClass<out
Annotation>)\n\n@Target(CLASS, PROPERTY, CONSTRUCTOR, FUNCTION,
TYPEALIAS)\n\n@Retention(BINARY)\n\ninternal annotation class WasExperimental(\n    vararg val markerClass:
KClass<out Annotation>)\n\n", "package kotlin\n\nimport kotlin.annotation.AnnotationTarget.*\n\n\n/**\n * This
annotation marks the standard library API that is considered experimental and is not subject to the\n * [general
compatibility guarantees](https://kotlinlang.org/docs/reference/evolution/components-stability.html) given for the
standard library:\n * the behavior of such API may be changed or the API may be removed completely in any
further release.\n *\n * > Beware using the annotated API especially if you're developing a library, since your library
might become binary incompatible\n * with the future versions of the standard library.\n *\n * Any usage of a
declaration annotated with `@ExperimentalStdlibApi` must be accepted either by\n * annotating that usage with the
[OptIn] annotation, e.g. `@OptIn(ExperimentalStdlibApi::class)`,\n * or by using the compiler argument `~opt-
in=kotlin.ExperimentalStdlibApi`. \n */\n\n@RequiresOptIn(level =

```

```

RequiresOptIn.Level.ERROR)\n@Retention(AnnotationRetention.BINARY)\n@Target(\n CLASS,\n ANNOTATION_CLASS,\n PROPERTY,\n FIELD,\n LOCAL_VARIABLE,\n VALUE_PARAMETER,\n CONSTRUCTOR,\n FUNCTION,\n PROPERTY_GETTER,\n PROPERTY_SETTER,\n TYPEALIAS)\n)\n@MustBeDocumented\n@SinceKotlin("1.3")\npublic annotation class
ExperimentalStdlibApi\n",/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\npackage kotlin\n\nimport kotlin.annotation.AnnotationTarget.*\nimport
kotlin.experimental.ExperimentalTypeInference\n\n/**\n * Allows to infer generic type arguments of a function
from the calls in the annotated function parameter of that function.\n *\n * When this annotation is placed on a
generic function parameter of a function,\n * it enables to infer the type arguments of that generic function from the
lambda body passed to that parameter.\n *\n * The calls that affect inference are either members of the receiver type
of an annotated function parameter or\n * extensions for that type. The extensions must be themselves annotated
with `@BuilderInference`.\n *\n * Example: we declare\n * ```\n * fun <T> sequence(@BuilderInference block:
suspend SequenceScope<T>().->Unit): Sequence<T>\n * ```\n * and use it like\n * ```\n * val result = sequence {
yield("result") }\n * ```\n * Here the type argument of the resulting sequence is inferred to `String` from\n * the
argument of the [SequenceScope.yield] function, that is called inside the lambda passed to [sequence].\n *\n * Note:
this annotation is experimental, see [ExperimentalTypeInference] on how to opt-in for it.\n
*/\n@Target(VALUE_PARAMETER, FUNCTION,
PROPERTY)\n@Retention(AnnotationRetention.BINARY)\n@SinceKotlin("1.3")\n@ExperimentalTypeInferenc
e\npublic annotation class BuilderInference\n\n/**\n * Enables overload selection based on the type of the value
returned from lambda argument.\n *\n * When two or more function overloads have otherwise the same parameter
lists that differ only in the return type\n * of a functional parameter, this annotation enables overload selection by the
type of the value returned from\n * the lambda function passed to this functional parameter.\n *\n * Example:\n *
```
\n * @OverloadResolutionByLambdaReturnType\n * fun create(intProducer: () -> Int): Int\n *\n * fun
create(doubleProducer: () -> Double): Double\n *\n * val newValue = create { 3.14 }\n * ```\n *\n * The annotation
being applied to one of overloads allows to resolve this ambiguity by analyzing what value is returned\n * from the
lambda function.\n *\n * This annotation is also used to discriminate the annotated overloads in case if overload
selection still cannot\n * choose one of them even taking in account the result of lambda parameter analysis. In that
case a warning is reported.\n *\n * Note: this annotation is experimental, see [ExperimentalTypeInference] on how
to opt-in for it.\n
*/\n@Target(FUNCTION)\n@Retention(AnnotationRetention.BINARY)\n@SinceKotlin("1.4")\n@Experimental
TypeInference\npublic annotation class OverloadResolutionByLambdaReturnType",/*\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\npackage kotlin\n\nimport
kotlin.annotation.AnnotationTarget.*\nimport kotlin.internal.RequireKotlin\nimport
kotlin.internal.RequireKotlinVersionKind\n\n/**\n * The experimental multiplatform support API marker.\n *\n * Any usage of a declaration annotated with `@ExperimentalMultiplatform` must be accepted either by\n * annotating
that usage with the [OptIn] annotation, e.g. `@OptIn(ExperimentalMultiplatform::class)`,\n * or by using the
compiler argument ` -opt-in=kotlin.ExperimentalMultiplatform `.\n
*/\n@RequiresOptIn\n@MustBeDocumented\n@Target(\n CLASS,\n ANNOTATION_CLASS,\n
PROPERTY,\n FIELD,\n LOCAL_VARIABLE,\n VALUE_PARAMETER,\n CONSTRUCTOR,\n
FUNCTION,\n PROPERTY_GETTER,\n PROPERTY_SETTER,\n
TYPEALIAS)\n)\n@Retention(AnnotationRetention.BINARY)\npublic annotation class
ExperimentalMultiplatform\n\n/**\n * Marks an expected annotation class that it isn't required to have actual
counterparts in all platforms.\n *\n * This annotation is only applicable to `expect` annotation classes in multi-
platform projects and marks that class as `optional`.\n *\n * Optional expected class is allowed to have no
corresponding actual class on the platform. Optional annotations can only be used\n * to annotate something, not as
types in signatures. If an optional annotation has no corresponding actual class on a platform,\n * the annotation

```

entries where it's used are simply erased when compiling code on that platform.

`* Note: this annotation is experimental, see [ExperimentalMultiplatform] on how to opt-in for it.`

```

\/n@Target(ANNOTATION_CLASS)\n@Retention(AnnotationRetention.BINARY)\n@ExperimentalMultiplatform\n\npublic annotation class OptionalExpectation\n"/\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\nimport kotlin.annotation.AnnotationRetention.BINARY\nimport kotlin.annotation.AnnotationRetention.SOURCE\nimport kotlin.annotation.AnnotationTarget.*\nimport kotlin.internal.RequireKotlin\nimport kotlin.internal.RequireKotlinVersionKind\nimport kotlin.reflect.KClass\n\n/**\n * Signals that the annotated annotation class is a marker of an API that requires an explicit opt-in.\n * Call sites of any declaration annotated with that marker should opt in to the API either by using [OptIn],\n * or by being annotated with that marker themselves, effectively causing further propagation of the opt-in requirement.\n * @property message message to be reported on usages of API without an explicit opt-in, or empty string for the default message.\n * The default message is: \"This declaration is experimental and its usage should be marked with 'Marker'\n * or '@OptIn(Marker::class)\", where `Marker` is the opt-in requirement marker.\n * @property level specifies how usages of API without an explicit opt-in are reported in code.\n */\n@Target(ANNOTATION_CLASS)\n@Retention(BINARY)\n@SinceKotlin(\"1.3\")\npublic annotation class RequiresOptIn(\n val message: String = \"\",\n val level: Level = Level.ERROR)\n\n/**\n * Severity of the diagnostic that should be reported on usages which did not explicitly opt into\n * the API either by using [OptIn] or by being annotated with the corresponding marker annotation.\n */\npublic enum class Level {\n /**\n * Specifies that a warning should be reported on incorrect usages of this API.\n */\n WARNING,\n /**\n * Specifies that an error should be reported on incorrect usages of this API.\n */\n ERROR,\n}\n\n/**\n * Allows to use the API denoted by the given markers in the annotated file, declaration, or expression.\n * If a declaration is annotated with [OptIn], its usages are **not** required to opt in to that API.\n */\n@Target(\n CLASS, PROPERTY, LOCAL_VARIABLE, VALUE_PARAMETER, CONSTRUCTOR, FUNCTION,\n PROPERTY_GETTER, PROPERTY_SETTER, EXPRESSION, FILE,\n TYPEALIAS)\n@Retention(SOURCE)\n@SinceKotlin(\"1.3\")\npublic annotation class OptIn(\n vararg val markerClass: KClass<out Annotation>)\n"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\nimport kotlin.js.JsName\n\n/**\n * Provides a skeletal implementation of the read-only [Collection] interface.\n */\n@param E the type of elements contained in the collection. The collection is covariant in its element type.\n@SinceKotlin(\"1.1\")\npublic abstract class AbstractCollection<out E> protected constructor() : Collection<E> {\n abstract override val size: Int\n abstract override fun iterator(): Iterator<E>\n\n override fun contains(element: @UnsafeVariance E): Boolean = any { it == element }\n\n override fun containsAll(elements: Collection<@UnsafeVariance E>): Boolean =\n elements.all { contains(it) } // use when js will support bound refs: elements.all(this::contains)\n\n override fun isEmpty(): Boolean = size == 0\n\n override fun toString(): String = joinToString(\",\", \"\", \"[\", \"]\")\n\n if (it === this) \"(this Collection)\" else it.toString()\n}\n\n/**\n * Returns new array of type `Array<Any?>` with the elements of this collection.\n */\n@JsName(\"toArray\")\nprotected open fun toArray(): Array<Any?> = copyToArrayImpl(this)\n\n/**\n * Fills the provided [array] or creates new array of the same type\n * and fills it with the elements of this collection.\n */\nprotected open fun <T> toArray(array: Array<T>): Array<T> = copyToArrayImpl(this, array)\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\nprivate enum class State {\n Ready,\n NotReady,\n Done,\n Failed\n}\n\n/**\n * A base class to simplify implementing iterators so that implementations only have to implement [computeNext]\n * to implement the iterator, calling [done] when the iteration is complete.\n */\npublic abstract class AbstractIterator<T> : Iterator<T> {\n private var state = State.NotReady\n private var nextValue: T? = null\n\n override fun hasNext(): Boolean {\n require(state !=

```

```

State.Failed)\n return when (state) {\n State.Done -> false\n State.Ready -> true\n else ->
tryToComputeNext()\n }\n }\n\n override fun next(): T {\n if (!hasNext()) throw
NoSuchElementException()\n state = State.NotReady\n @Suppress(\"UNCHECKED_CAST\")\n return nextValue as T\n }\n\n private fun tryToComputeNext(): Boolean {\n state = State.Failed\n computeNext()\n return state == State.Ready\n }\n\n /**\n * Computes the next item in the iterator.\n * This callback method should call one of these two methods:\n * [setNext] with the next value of the iteration\n * [done] to indicate there are no more elements\n * Failure to call either method will result in the iteration terminating with a failed state\n */\n abstract protected fun computeNext(): Unit\n\n /**\n * Sets the next value in the iteration, called from the [computeNext] function\n */\n protected fun setNext(value: T): Unit {\n nextValue = value\n state = State.Ready\n }\n\n /**\n * Sets the state to done so that the iteration terminates.\n */\n protected fun done() {\n state = State.Done\n }\n}\n\n\"/>\n\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n * Based on GWT AbstractList\n * Copyright 2007 Google Inc.\n */\n\npackage kotlin.collections\n\n * Provides a skeletal implementation of the read-only [List] interface.\n * This class is intended to help implementing read-only lists so it doesn't support concurrent modification tracking.\n * @param E the type of elements contained in the list. The list is covariant in its element type.\n */\n\n@SinceKotlin(\"1.1\")\npublic abstract class AbstractList<out E>\nprotected constructor() : AbstractCollection<E>(), List<E> {\n abstract override val size: Int\n abstract override fun get(index: Int): E\n override fun iterator(): Iterator<E> = IteratorImpl()\n override fun indexOf(element: @UnsafeVariance E): Int = indexOfFirst { it == element }\n override fun lastIndexOf(element: @UnsafeVariance E): Int = indexOfLast { it == element }\n override fun listIterator(): ListIterator<E> = ListIteratorImpl(0)\n override fun listIterator(index: Int): ListIterator<E> = ListIteratorImpl(index)\n override fun subList(fromIndex: Int, toIndex: Int): List<E> = SubList(this, fromIndex, toIndex)\n\n private class SubList<out E>(private val list: AbstractList<E>, private val fromIndex: Int, toIndex: Int) : AbstractList<E>(), RandomAccess {\n private var _size: Int = 0\n\n init {\n checkRangeIndexes(fromIndex, toIndex, list.size)\n this._size = toIndex - fromIndex\n }\n\n override fun get(index: Int): E {\n checkElementIndex(index, _size)\n return list[fromIndex + index]\n }\n\n override val size: Int\n get() = _size\n }\n\n /**\n * Compares this list with other list instance with the ordered structural equality.\n * @return true, if [other] instance is a [List] of the same size, which contains the same elements in the same order.\n */\n override fun equals(other: Any?): Boolean {\n if (other === this) return true\n if (other !is List<*>) return false\n return orderedEquals(this, other)\n }\n\n /**\n * Returns the hash code value for this list.\n */\n override fun hashCode(): Int = orderedHashCode(this)\n\n private open inner class IteratorImpl : Iterator<E> {\n /** the index of the item that will be returned on the next call to [next]`()`\n */\n protected var index = 0\n\n override fun hasNext(): Boolean = index < size\n\n override fun next(): E {\n if (!hasNext()) throw NoSuchElementException()\n return get(index++)\n }\n }\n\n /**\n * Implementation of [ListIterator] for abstract lists.\n */\n private open inner class ListIteratorImpl(index: Int) : IteratorImpl(), ListIterator<E> {\n init {\n checkPositionIndex(index, this@AbstractList.size)\n this.index = index\n }\n\n override fun hasPrevious(): Boolean = index > 0\n\n override fun nextIndex(): Int = index\n\n override fun previous(): E {\n if (!hasPrevious()) throw NoSuchElementException()\n return get(--index)\n }\n\n override fun previousIndex(): Int = index - 1\n }\n\n internal companion object {\n internal fun checkElementIndex(index: Int, size: Int) {\n if (index < 0 || index >= size) {\n throw IndexOutOfBoundsException(\"index: $index, size: $size\")\n }\n }\n\n internal fun checkPositionIndex(index: Int, size: Int) {\n if (index < 0 || index > size) {\n throw IndexOutOfBoundsException(\"index: $index, size: $size\")\n }\n }\n\n internal fun checkRangeIndexes(fromIndex: Int, toIndex: Int, size: Int) {\n if (fromIndex < 0 || toIndex > size) {\n throw IndexOutOfBoundsException(\"fromIndex: $fromIndex, toIndex: $toIndex, size: $size\")\n }\n if (fromIndex > toIndex) {\n throw IllegalArgumentException(\"fromIndex: $fromIndex > toIndex: $toIndex\")\n }\n }\n\n internal fun checkBoundsIndexes(startIndex: Int, endIndex: Int, size: Int) {\n

```

```

 if (startIndex < 0 || endIndex > size) {\n throw IndexOutOfBoundsException("startIndex:
$startIndex, endIndex: $endIndex, size: $size")\n }\n if (startIndex > endIndex) {\n throw
IllegalArgumentException("startIndex: $startIndex > endIndex: $endIndex")\n }\n }\n internal
fun orderedHashCode(c: Collection<*>): Int {\n var hashCode = 1\n for (e in c) {\n
hashCode = 31 * hashCode + (e?.hashCode() ?: 0)\n }\n return hashCode\n }\n internal fun
orderedEquals(c: Collection<*>, other: Collection<*>): Boolean {\n if (c.size != other.size) return false\n
 val otherIterator = other.iterator()\n for (elem in c) {\n val elemOther = otherIterator.next()\n
 if (elem != elemOther) {\n return false\n }\n return true\n }\n
 }\n}"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this
source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n
*\n * Based on GWT AbstractMap\n * Copyright 2007 Google Inc.\n */\n\npackage kotlin.collections\n\n/**\n * Provides
a skeletal implementation of the read-only [Map] interface.\n *\n * The implementor is required to implement
[entries] property, which should return read-only set of map entries.\n *\n * @param K the type of map keys. The
map is invariant in its key type.\n *\n * @param V the type of map values. The map is covariant in its value type.\n
*/\n\n@SinceKotlin("1.1")\npublic abstract class AbstractMap<K, out V> protected constructor() : Map<K, V>
{\n\n override fun containsKey(key: K): Boolean {\n return implFindEntry(key) != null\n }\n\n override
fun containsValue(value: @UnsafeVariance V): Boolean = entries.any { it.value == value }\n\n internal fun
containsEntry(entry: Map.Entry<*, *>?): Boolean {\n // since entry comes from @UnsafeVariance parameters it
can be virtually anything\n if (entry != Map.Entry<*, *>) return false\n val key = entry.key\n val value
= entry.value\n val ourValue = get(key)\n\n if (value != ourValue) {\n return false\n }\n\n //
Perhaps it was null and we don't contain the key?\n if (ourValue == null && !containsKey(key)) {\n
return false\n }\n\n return true\n }\n\n /**\n * Compares this map with other instance with the
ordered structural equality.\n *\n * @return true, if [other] instance is a [Map] of the same size, all entries of
which are contained in the [entries] set of this map.\n */\n\n override fun equals(other: Any?): Boolean {\n if
(other === this) return true\n if (other != Map<*, *>) return false\n if (size != other.size) return false\n
return other.entries.all { containsEntry(it) }\n }\n\n override operator fun get(key: K): V? =
implFindEntry(key)?.value\n\n /**\n * Returns the hash code value for this map.\n *\n * It is the same as
the hashCode of [entries] set.\n */\n\n override fun hashCode(): Int = entries.hashCode()\n\n override fun
isEmpty(): Boolean = size == 0\n\n override val size: Int get() = entries.size\n\n /**\n * Returns a read-only
[Set] of all keys in this map.\n *\n * Accessing this property first time creates a keys view from [entries].\n *
All subsequent accesses just return the created instance.\n */\n\n override val keys: Set<K>\n get() {\n
if (_keys == null) {\n _keys = object : AbstractSet<K>() {\n override operator fun
contains(element: K): Boolean = containsKey(element)\n\n override operator fun iterator(): Iterator<K>
{\n val entryIterator = entries.iterator()\n return object : Iterator<K> {\n
 override fun hasNext(): Boolean = entryIterator.hasNext()\n override fun
next(): K = entryIterator.next().key\n }\n }\n }\n return _keys!!\n }\n
 }\n\n @kotlin.jvm.Volatile\n private var _keys: Set<K>? = null\n\n override fun toString(): String = entries.joinToString(", ", "\n", "\n") {\n
toString(it) }\n\n private fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "\n" +
toString(entry.value)\n\n private fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()\n\n
 /**\n * Returns a read-only [Collection] of all values in this map.\n *\n * Accessing this property first time
creates a values view from [entries].\n *\n * All subsequent accesses just return the created instance.\n */\n
 override val values: Collection<V>\n get() {\n if (_values == null) {\n _values = object :
AbstractCollection<V>() {\n override operator fun contains(element: @UnsafeVariance V): Boolean =
containsValue(element)\n\n override operator fun iterator(): Iterator<V> {\n val
entryIterator = entries.iterator()\n return object : Iterator<V> {\n override fun
hasNext(): Boolean = entryIterator.hasNext()\n override fun next(): V = entryIterator.next().value\n
 }\n }\n }\n }\n return _values\n }\n\n override val size: Int get() = this@AbstractMap.size\n}

```



```

 }\n return _values!!\n }\n\n @kotlin.jvm.Volatile\n private var _values: Collection<V>? = null\n\n private fun implFindEntry(key: K): Map.Entry<K, V>? = entries.firstOrNull { it.key == key }\n\n internal\n companion object {\n internal fun entryHashCode(e: Map.Entry<*, *>): Int = with(e) { (key?.hashCode() ?: 0) xor (value?.hashCode() ?: 0) }\n internal fun entryToString(e: Map.Entry<*, *>): String = with(e) {\n \"$key=$value\" }\n internal fun entryEquals(e: Map.Entry<*, *>, other: Any?): Boolean {\n if (other !is Map.Entry<*, *>) return false\n return e.key == other.key && e.value == other.value\n }\n }\n\n}\n}\n\n"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\npackage kotlin.collections\n/**\n * Provides a skeletal implementation of the read-only [Set] interface.\n *\n * This class is intended to help implementing read-only sets so it doesn't support concurrent modification tracking.\n *\n * @param E the type of elements contained in the set. The set is covariant in its element type.\n */\n@SinceKotlin("1.1")\npublic abstract class AbstractSet<out E> protected constructor() : AbstractCollection<E>(), Set<E> {\n /**\n * Compares this set with other set instance with the unordered structural equality.\n *\n * @return true, if [other] instance is a [Set] of the same size, all elements of which are contained in this set.\n *\n * override fun equals(other: Any?): Boolean {\n * if (other === this) return true\n * if (other !is Set<*>) return false\n * return setEquals(this, other)\n * }\n */\n /**\n * Returns the hash code value for this set.\n *\n * override fun hashCode(): Int = unorderedHashCode(this)\n */\n internal companion object {\n internal fun unorderedHashCode(c: Collection<*>): Int {\n var hashCode = 0\n for (element in c) {\n hashCode += (element?.hashCode() ?: 0)\n }\n return hashCode\n }\n internal fun setEquals(c: Set<*>, other: Set<*>): Boolean {\n if (c.size != other.size) return false\n return c.containsAll(other)\n }\n }\n}\n}\n\n"/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\npackage kotlin.collections\n/**\n * Resizable-array implementation of the deque data structure.\n *\n * The name deque is short for "double ended queue" and is usually pronounced "deck".\n *\n * The collection provide methods for convenient access to the both ends.\n *\n * It also implements [MutableList] interface and supports efficient get/set operations by index.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic class ArrayDeque<E> : AbstractMutableList<E> {\n private var head: Int = 0\n private var elementData: Array<Any?>\n\n override var size: Int = 0\n private set\n\n /**\n * Constructs an empty deque with specified [initialCapacity], or throws [IllegalArgumentException] if [initialCapacity] is negative.\n *\n * public constructor(initialCapacity: Int) {\n * elementData = when {\n * initialCapacity == 0 -> emptyElementData\n * initialCapacity > 0 -> arrayOfNulls(initialCapacity)\n * }\n * else -> throw IllegalArgumentException("Illegal Capacity: $initialCapacity")\n * }\n */\n /**\n * Constructs an empty deque.\n *\n * public constructor() {\n * elementData = emptyElementData\n * }\n */\n /**\n * Constructs a deque that contains the same elements as the specified [elements] collection in the same order.\n *\n * public constructor(elements: Collection<E>) {\n * elementData = elements.toArray()\n * size = elementData.size\n * if (elementData.isEmpty())\n * elementData = emptyElementData\n * }\n */\n /**\n * Ensures that the capacity of this deque is at least equal to the specified [minCapacity].\n *\n * If the current capacity is less than the [minCapacity], a new backing storage is allocated with greater capacity.\n *\n * Otherwise, this method takes no action and simply returns.\n *\n * private fun ensureCapacity(minCapacity: Int) {\n * if (minCapacity < 0) throw IllegalStateException("Deque is too big.") // overflow\n * if (minCapacity <= elementData.size) return\n * if (elementData === emptyElementData) {\n * elementData = arrayOfNulls(minCapacity.coerceAtLeast(defaultMinCapacity))\n * }\n * return\n * }\n */\n val newCapacity = newCapacity(elementData.size, minCapacity)\n copyElements(newCapacity)\n}\n\n/**\n * Creates a new array with the specified [newCapacity] size and copies elements in the [elementData] array to it.\n *\n * private fun copyElements(newCapacity: Int) {\n val newElements = arrayOfNulls<Any?>(newCapacity)\n elementData.copyInto(newElements, 0, head, elementData.size)\n elementData.copyInto(newElements, elementData.size - head, 0, head)\n head = 0\n elementData = newElements\n}\n}\n\n@kotlin.internal.InlineOnly\nprivate inline fun

```

```

internalGet(internalIndex: Int): E {
 @Suppress("UNCHECKED_CAST")
 return
 elementData[internalIndex] as E
}

private fun positiveMod(index: Int): Int = if (index >=
elementData.size) index - elementData.size else index

private fun negativeMod(index: Int): Int = if (index < 0)
index + elementData.size else index

@kotlin.internal.InlineOnly
private inline fun internalIndex(index:
Int): Int = positiveMod(head + index)

private fun incremented(index: Int): Int = if (index ==
elementData.lastIndex) 0 else index + 1

private fun decremented(index: Int): Int = if (index == 0)
elementData.lastIndex else index - 1

override fun isEmpty(): Boolean = size == 0

/**
 * Returns the
first element, or throws [NoSuchElementException] if this deque is empty.
*/
public fun first(): E = if
(isEmpty()) throw NoSuchElementException("ArrayDeque is empty.")
else internalGet(head)

/**
 * Returns the first element, or `null` if this deque is empty.
*/
public fun firstOrNull(): E? = if (isEmpty()) null
else internalGet(head)

/**
 * Returns the last element, or throws [NoSuchElementException] if this deque
is empty.
*/
public fun last(): E = if (isEmpty()) throw NoSuchElementException("ArrayDeque is empty.")
else internalGet(internalIndex(lastIndex))

/**
 * Returns the last element, or `null` if this deque is empty.
*/
public fun lastOrNull(): E? = if (isEmpty()) null
else internalGet(internalIndex(lastIndex))

Prepends the specified [element] to this deque.
*/
public fun addFirst(element: E) {
 ensureCapacity(size + 1)
 head = decremented(head)
 elementData[head] = element
 size += 1
}

/**
 * Appends the specified [element] to this deque.
*/
public fun addLast(element: E) {
 ensureCapacity(size + 1)
 elementData[internalIndex(size)] = element
 size += 1
}

Removes the first element from this deque and returns that removed element, or throws [NoSuchElementException]
if this deque is empty.
*/
public fun removeFirst(): E {
 if (isEmpty()) throw
NoSuchElementException("ArrayDeque is empty.")
 val element = internalGet(head)
 elementData[head] = null
 head = incremented(head)
 size -= 1
 return element
}

/**
 * Removes the first element from this deque and returns that removed element, or returns `null` if this deque is
empty.
*/
public fun removeFirstOrNull(): E? = if (isEmpty()) null
else removeFirst()

Removes the last element from this deque and returns that removed element, or throws [NoSuchElementException]
if this deque is empty.
*/
public fun removeLast(): E {
 if (isEmpty()) throw
NoSuchElementException("ArrayDeque is empty.")
 val internalLastIndex = internalIndex(lastIndex)
 val element = internalGet(internalLastIndex)
 elementData[internalLastIndex] = null
 size -= 1
 return element
}

/**
 * Removes the last element from this deque and returns that removed element, or
returns `null` if this deque is empty.
*/
public fun removeLastOrNull(): E? = if (isEmpty()) null
else
removeLast()

// MutableList, MutableCollection
public override fun add(element: E): Boolean {
 addLast(element)
 return true
}

public override fun add(index: Int, element: E) {
 AbstractList.checkPositionIndex(index, size)
 if (index == size) {
 addLast(element)
 } else if (index == 0) {
 addFirst(element)
 }
 ensureCapacity(size
+ 1)

// Elements in circular array lay in 2 ways:
// 1. `head` is less than `tail`: [#, #, e1, e2, e3,
#]
// 2. `head` is greater than `tail`: [e3, #, #, #, e1, e2]
// where head is the index of the first element
in the circular array,
// and tail is the index following the last element.
// At this point the
insertion index is not equal to head or tail.
// Also the circular array can store at least one more element.
// Depending on where the given element must be inserted the preceding or the succeeding
// elements
will be shifted to make room for the element to be inserted.
// In case the preceding elements are
shifted:
// * if the insertion index is greater than the head (regardless of circular array form)
// ->
shift the preceding elements
// * otherwise, the circular array has (2) form and the insertion index is less than
tail
// -> shift all elements in the back of the array
// -> shift preceding elements in the front of the
array
// In case the succeeding elements are shifted:
// * if the insertion index is less than the tail
(regardless of circular array form)
// -> shift the succeeding elements
// * otherwise, the circular
array has (2) form and the insertion index is greater than head
// -> shift all elements in the front of the
array
// -> shift succeeding elements in the back of the array
 val internalIndex =
internalIndex(index)
 if (index < (size + 1) shr 1) {
 // closer to the first element -> shift preceding

```

```

elements\n val decrementedInternalIndex = decremented(internalIndex)\n val decrementedHead =
decremented(head)\n if (decrementedInternalIndex >= head) {\n elementData[decrementedHead]
= elementData[head] // head can be zero\n elementData.copyInto(elementData, head, head + 1,
decrementedInternalIndex + 1)\n } else { // head > tail\n elementData.copyInto(elementData, head -
1, head, elementData.size) // head can't be zero\n elementData[elementData.size - 1] = elementData[0]\n
 elementData.copyInto(elementData, 0, 1, decrementedInternalIndex + 1)\n }\n\n
elementData[decrementedInternalIndex] = element\n head = decrementedHead\n } else {\n //
closer to the last element -> shift succeeding elements\n val tail = internalIndex(size)\n\n if
(internalIndex < tail) {\n elementData.copyInto(elementData, internalIndex + 1, internalIndex, tail)\n
} else { // head > tail\n elementData.copyInto(elementData, 1, 0, tail)\n elementData[0] =
elementData[elementData.size - 1]\n elementData.copyInto(elementData, internalIndex + 1, internalIndex,
elementData.size - 1)\n }\n\n elementData[internalIndex] = element\n }\n size += 1\n }\n\n
private fun copyCollectionElements(internalIndex: Int, elements: Collection<E>) {\n val iterator =
elements.iterator()\n for (index in internalIndex until elementData.size) {\n if (!iterator.hasNext())
break\n elementData[index] = iterator.next()\n }\n for (index in 0 until head) {\n if
(!iterator.hasNext()) break\n elementData[index] = iterator.next()\n }\n\n size += elements.size\n
}\n\n public override fun addAll(elements: Collection<E>): Boolean {\n if (elements.isEmpty()) return false\n
 ensureCapacity(this.size + elements.size)\n copyCollectionElements(internalIndex(size), elements)\n
 return true\n }\n\n public override fun addAll(index: Int, elements: Collection<E>): Boolean {\n
 AbstractList.checkPositionIndex(index, size)\n if (elements.isEmpty()) {\n return false\n } else if
(index == size) {\n return addAll(elements)\n }\n\n ensureCapacity(this.size + elements.size)\n\n
 val tail = internalIndex(size)\n val internalIndex = internalIndex(index)\n val elementsSize =
elements.size\n\n if (index < (size + 1) shr 1) {\n // closer to the first element -> shift preceding
elements\n\n var shiftedHead = head - elementsSize\n\n if (internalIndex >= head) {\n if
(shiftedHead >= 0) {\n elementData.copyInto(elementData, shiftedHead, head, internalIndex)\n
} else { // head < tail, insertion leads to head >= tail\n shiftedHead += elementData.size\n val
elementsToShift = internalIndex - head\n val shiftToBack = elementData.size - shiftedHead\n\n if
(shiftToBack >= elementsToShift) {\n elementData.copyInto(elementData, shiftedHead, head,
internalIndex)\n } else {\n elementData.copyInto(elementData, shiftedHead, head, head +
shiftToBack)\n elementData.copyInto(elementData, 0, head + shiftToBack, internalIndex)\n
}\n }\n } else { // head > tail, internalIndex < tail\n elementData.copyInto(elementData,
shiftedHead, head, elementData.size)\n if (elementsSize >= internalIndex) {\n
elementData.copyInto(elementData, elementData.size - elementsSize, 0, internalIndex)\n } else {\n
 elementData.copyInto(elementData, elementData.size - elementsSize, 0, elementsSize)\n
 elementData.copyInto(elementData, 0, elementsSize, internalIndex)\n }\n }\n head =
shiftedHead\n copyCollectionElements(negativeMod(internalIndex - elementsSize), elements)\n } else
{\n // closer to the last element -> shift succeeding elements\n\n val shiftedInternalIndex =
internalIndex + elementsSize\n\n if (internalIndex < tail) {\n if (tail + elementsSize <=
elementData.size) {\n elementData.copyInto(elementData, shiftedInternalIndex, internalIndex, tail)\n
} else { // head < tail, insertion leads to head >= tail\n if (shiftedInternalIndex >= elementData.size)
{\n elementData.copyInto(elementData, shiftedInternalIndex - elementData.size, internalIndex, tail)\n
}\n } else {\n val shiftToFront = tail + elementsSize - elementData.size\n
elementData.copyInto(elementData, 0, tail - shiftToFront, tail)\n elementData.copyInto(elementData,
shiftedInternalIndex, internalIndex, tail - shiftToFront)\n }\n }\n } else { // head > tail,
internalIndex > head\n elementData.copyInto(elementData, elementsSize, 0, tail)\n if
(shiftedInternalIndex >= elementData.size) {\n elementData.copyInto(elementData, shiftedInternalIndex
- elementData.size, internalIndex, elementData.size)\n } else {\n
elementData.copyInto(elementData, 0, elementData.size - elementsSize, elementData.size)\n
}

```

```

elementData.copyInto(elementData, shiftedInternalIndex, internalIndex, elementData.size - elementsSize)\n
}\n }\n copyCollectionElements(internalIndex, elements)\n }\n\n return true\n }\n\n public
override fun get(index: Int): E {\n AbstractList.checkElementIndex(index, size)\n\n return
internalGet(internalIndex(index))\n }\n\n public override fun set(index: Int, element: E): E {\n
AbstractList.checkElementIndex(index, size)\n\n val internalIndex = internalIndex(index)\n val oldElement
= internalGet(internalIndex)\n elementData[internalIndex] = element\n\n return oldElement\n }\n\n
public override fun contains(element: E): Boolean = indexOf(element) != -1\n\n public override fun
indexOf(element: E): Int {\n val tail = internalIndex(size)\n\n if (head < tail) {\n for (index in head
until tail) {\n if (element == elementData[index]) return index - head\n }\n } else if (head >=
tail) {\n for (index in head until elementData.size) {\n if (element == elementData[index]) return
index - head\n }\n for (index in 0 until tail) {\n if (element == elementData[index]) return
index + elementData.size - head\n }\n }\n\n return -1\n }\n\n public override fun
lastIndexOf(element: E): Int {\n val tail = internalIndex(size)\n\n if (head < tail) {\n for (index in tail
- 1 downTo head) {\n if (element == elementData[index]) return index - head\n }\n } else if
(head > tail) {\n for (index in tail - 1 downTo 0) {\n if (element == elementData[index]) return
index + elementData.size - head\n }\n for (index in elementData.lastIndex downTo head) {\n
if (element == elementData[index]) return index - head\n }\n }\n\n return -1\n }\n\n public
override fun remove(element: E): Boolean {\n val index = indexOf(element)\n if (index == -1) return
false\n removeAt(index)\n return true\n }\n\n public override fun removeAt(index: Int): E {\n
AbstractList.checkElementIndex(index, size)\n\n if (index == lastIndex) {\n return removeLast()\n }
else if (index == 0) {\n return removeFirst()\n }\n\n val internalIndex = internalIndex(index)\n
val element = internalGet(internalIndex)\n\n if (index < size shr 1) {\n // closer to the first element ->
shift preceding elements\n if (internalIndex >= head) {\n elementData.copyInto(elementData, head
+ 1, head, internalIndex)\n } else { // head > tail, internalIndex < head\n
elementData.copyInto(elementData, 1, 0, internalIndex)\n elementData[0] = elementData[elementData.size
- 1]\n elementData.copyInto(elementData, head + 1, head, elementData.size - 1)\n }\n\n
elementData[head] = null\n head = incremented(head)\n } else {\n // closer to the last element ->
shift succeeding elements\n val internalLastIndex = internalIndex(lastIndex)\n\n if (internalIndex <=
internalLastIndex) {\n elementData.copyInto(elementData, internalIndex, internalIndex + 1,
internalLastIndex + 1)\n } else { // head > tail, internalIndex > head\n
elementData.copyInto(elementData, internalIndex, internalIndex + 1, elementData.size)\n
elementData[elementData.size - 1] = elementData[0]\n elementData.copyInto(elementData, 0, 1,
internalLastIndex + 1)\n }\n\n elementData[internalLastIndex] = null\n }\n size -= 1\n\n
return element\n }\n\n public override fun removeAll(elements: Collection<E>): Boolean = filterInPlace {
!elements.contains(it) }\n\n public override fun retainAll(elements: Collection<E>): Boolean = filterInPlace {
elements.contains(it) }\n\n private inline fun filterInPlace(predicate: (E) -> Boolean): Boolean {\n if
(this.isEmpty() || elementData.isEmpty())\n return false\n\n val tail = internalIndex(size)\n var
newTail = head\n var modified = false\n\n if (head < tail) {\n for (index in head until tail) {\n
val element = elementData[index]\n\n @Suppress("UNCHECKED_CAST")\n if
(predicate(element as E))\n elementData[newTail++] = element\n else\n modified =
true\n }\n\n elementData.fill(null, newTail, tail)\n } else {\n for (index in head until
elementData.size) {\n val element = elementData[index]\n elementData[index] = null\n\n
@Suppress("UNCHECKED_CAST")\n if (predicate(element as E))\n elementData[newTail++] = element\n else\n modified = true\n }\n\n newTail =
positiveMod(newTail)\n\n for (index in 0 until tail) {\n val element = elementData[index]\n
elementData[index] = null\n\n @Suppress("UNCHECKED_CAST")\n if (predicate(element as
E)) {\n elementData[newTail] = element\n newTail = incremented(newTail)\n }\n
else {\n modified = true\n }\n }\n }\n if (modified)\n size =

```

```

negativeMod(newTail - head)\n\n return modified\n }\n\n public override fun clear() {\n val tail =
internalIndex(size)\n if (head < tail) {\n elementData.fill(null, head, tail)\n } else if (isEmpty())
{\n elementData.fill(null, head, elementData.size)\n elementData.fill(null, 0, tail)\n }\n head =
0\n size = 0\n }\n\n @Suppress("NOTHING_TO_OVERRIDE")\n override fun <T> toArray(array:
Array<T>): Array<T> {\n @Suppress("UNCHECKED_CAST")\n val dest = (if (array.size >= size) array
else arrayOfNulls(array, size)) as Array<Any?>\n val tail = internalIndex(size)\n if (head < tail) {\n
elementData.copyInto(dest, startIndex = head, endIndex = tail)\n } else if (isEmpty()) {\n
elementData.copyInto(dest, destinationOffset = 0, startIndex = head, endIndex = elementData.size)\n
elementData.copyInto(dest, destinationOffset = elementData.size - head, startIndex = 0, endIndex = tail)\n }\n
 if (dest.size > size) {\n dest[size] = null // null-terminate\n }\n\n @Suppress("UNCHECKED_CAST")\n return dest as Array<T>\n }\n\n @Suppress("NOTHING_TO_OVERRIDE")\n override fun toArray(): Array<Any?> {\n return
toArray(arrayOfNulls<Any?>(size))\n }\n\n // for testing\n internal fun <T> testToArray(array: Array<T>):
Array<T> = toArray(array)\n internal fun testToArray(): Array<Any?> = toArray()\n\n internal companion
object {\n private val emptyElementData = emptyArray<Any?>()\n private const val maxArraySize =
Int.MAX_VALUE - 8\n private const val defaultMinCapacity = 10\n internal fun
newCapacity(oldCapacity: Int, minCapacity: Int): Int {\n // overflow-conscious\n var newCapacity =
oldCapacity + (oldCapacity shr 1)\n if (newCapacity - minCapacity < 0)\n newCapacity =
minCapacity\n if (newCapacity - maxArraySize > 0)\n newCapacity = if (minCapacity >
maxArraySize) Int.MAX_VALUE else maxArraySize\n return newCapacity\n }\n }\n\n // For testing
only\n internal fun internalStructure(structure: (head: Int, elements: Array<Any?>) -> Unit) {\n val tail =
internalIndex(size)\n val head = if (isEmpty() || head < tail) head else head - elementData.size\n
structure(head, toArray())\n }\n}"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("ArraysKt")\n\npackage
kotlin.collections\nimport kotlin.contracts.*\n\n/**\n * Returns a single list of all elements from all arrays in the
given array.\n * @sample samples.collections.Arrays.Transformations.flattenArray\n */\npublic fun <T> Array<out
T>.flatten(): List<T> {\n val result = ArrayList<T>(sumOf { it.size })\n for (element in this) {\n
result.addAll(element)\n }\n return result\n}\n\n/**\n * Returns a pair of lists, where\n * *first* list is built from
the first values of each pair from this array,\n * *second* list is built from the second values of each pair from this
array.\n * @sample samples.collections.Arrays.Transformations.unzipArray\n */\npublic fun <T, R> Array<out
Pair<T, R>>.unzip(): Pair<List<T>, List<R>> {\n val listT = ArrayList<T>(size)\n val listR =
ArrayList<R>(size)\n for (pair in this) {\n listT.add(pair.first)\n listR.add(pair.second)\n }\n return
listT to listR\n}\n\n/**\n * Returns `true` if this nullable array is either null or empty.\n * @sample
samples.collections.Arrays.Usage.arrayIsNullOrEmpty\n
/\n\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun Array<>.isNullOrEmpty(): Boolean
{\n contract {\n returns(false) implies (this@isNullOrEmpty != null)\n }\n return this == null ||
this.isEmpty()\n}\n\n/**\n * Returns this array if it's not empty\n * or the result of calling [defaultValue] function if
the array is empty.\n * @sample samples.collections.Arrays.Usage.arrayIfEmpty\n
/\n\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("UPPER_BOUND_CANNOT_BE_ARRAY")\npublic inline fun <C, R> C.ifEmpty(defaultValue: () -> R): R where C : Array<>, C : R =\n if (isEmpty())
defaultValue() else
this\n\n\n@OptIn(ExperimentalUnsignedTypes::class)\n@SinceKotlin("1.3")\n@PublishedApi\n@kotlin.jvm.Jvm
Name("contentDeepEquals")\n@kotlin.js.JsName("contentDeepEqualsImpl")\ninternal fun <T> Array<out
T>?.contentDeepEqualsImpl(other: Array<out T>?): Boolean {\n if (this === other) return true\n if (this == null
|| other == null || this.size != other.size) return false\n for (i in indices) {\n val v1 = this[i]\n val v2 =
other[i]\n if (v1 === v2) {\n continue\n } else if (v1 == null || v2 == null) {\n return false\n

```

```

}
when {
 v1 is Array<*> && v2 is Array<*> -> if (!v1.contentDeepEquals(v2)) return false
 v1 is ByteArray && v2 is ByteArray -> if (!v1.contentEquals(v2)) return false
 v1 is ShortArray && v2 is ShortArray -> if (!v1.contentEquals(v2)) return false
 v1 is IntArray && v2 is IntArray -> if (!v1.contentEquals(v2)) return false
 v1 is LongArray && v2 is LongArray -> if (!v1.contentEquals(v2)) return false
 v1 is FloatArray && v2 is FloatArray -> if (!v1.contentEquals(v2)) return false
 v1 is DoubleArray && v2 is DoubleArray -> if (!v1.contentEquals(v2)) return false
 v1 is CharArray && v2 is CharArray -> if (!v1.contentEquals(v2)) return false
 v1 is BooleanArray && v2 is BooleanArray -> if (!v1.contentEquals(v2)) return false
 v1 is UByteArray && v2 is UByteArray -> if (!v1.contentEquals(v2)) return false
 v1 is UShortArray && v2 is UShortArray -> if (!v1.contentEquals(v2)) return false
 v1 is UIntArray && v2 is UIntArray -> if (!v1.contentEquals(v2)) return false
 v1 is ULongArray && v2 is ULongArray -> if (!v1.contentEquals(v2)) return false
} else -> if (v1 != v2) return false
}
return true
}
@SinceKotlin("1.3")
@PublishedApi
@kotlin.jvm.JvmName("contentDeepToString")
@kotlin.js.JsName("contentDeepToStringImpl")
internal fun <T> Array<out T>?.contentDeepToStringImpl(): String {
 if (this == null) return "null"
 val length = size.coerceAtMost((Int.MAX_VALUE - 2) / 5) * 5 + 2 // in order not to overflow Int.MAX_VALUE
 return buildString(length) {
 contentDeepToStringInternal(this, mutableListOf())
 }
}
@OptIn(ExperimentalUnsignedTypes::class)
private fun <T> Array<out T>.contentDeepToStringInternal(result: StringBuilder, processed: MutableList<Array<*>>) {
 if (this in processed) {
 result.append("[...]")
 return
 }
 processed.add(this)
 result.append("[")
 for (i in indices) {
 if (i != 0) {
 result.append(", ")
 }
 val element = this[i]
 when (element) {
 null -> result.append("null")
 is Array<*> -> element.contentDeepToStringInternal(result, processed)
 is ByteArray -> result.append(element.contentToString())
 is ShortArray -> result.append(element.contentToString())
 is IntArray -> result.append(element.contentToString())
 is LongArray -> result.append(element.contentToString())
 is FloatArray -> result.append(element.contentToString())
 is DoubleArray -> result.append(element.contentToString())
 is CharArray -> result.append(element.contentToString())
 is BooleanArray -> result.append(element.contentToString())
 is UByteArray -> result.append(element.contentToString())
 is UShortArray -> result.append(element.contentToString())
 is UIntArray -> result.append(element.contentToString())
 is ULongArray -> result.append(element.contentToString())
 else -> result.append(element.toString())
 }
 }
 result.append("]")
 processed.removeAt(processed.lastIndex)
}
/* Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
package kotlin.collections
/** Returns true if the brittle contains optimization is enabled. See KT-45438.
 */
internal expect fun brittleContainsOptimizationEnabled(): Boolean
/** Returns true if [brittleContainsOptimizationEnabled] is true
 * and it's safe to convert this collection to a set without changing contains method behavior.
 */
private fun <T> Collection<T>.safeToConvertToSet() = brittleContainsOptimizationEnabled() && size > 2 && this is ArrayList
/** When [brittleContainsOptimizationEnabled] is true:
 * - Converts this [Iterable] to a set if it is not a [Collection].
 * - Converts this [Collection] to a set, when it's worth so and it doesn't change contains method behavior.
 * - Otherwise returns this.
 */
/** When [brittleContainsOptimizationEnabled] is false:
 * - Converts this [Iterable] to a list if it is not a [Collection].
 * - Otherwise returns this.
 */
internal fun <T> Iterable<T>.convertToSetForSetOperationWith(source: Iterable<T>): Collection<T> =
 when (this) {
 is Set -> this
 is Collection -> when {
 source is Collection && source.size < 2 -> this
 else -> if (this.safeToConvertToSet()) toHashSet() else this
 }
 else -> if (brittleContainsOptimizationEnabled()) toHashSet() else toList()
 }
/** When [brittleContainsOptimizationEnabled] is true:
 * - Converts this [Iterable] to a set if it is not a [Collection].
 * - Converts this [Collection] to a set, when it's worth so and it doesn't change contains method behavior.
 */

```

```

Otherwise returns this.\n * When [brittleContainsOptimizationEnabled] is false:\n * - Converts this [Iterable] to a list if it is not a [Collection].\n * - Otherwise returns this.\n *^\ninternal fun <T>
Iterable<T>.convertToSetForSetOperation(): Collection<T> =\n when (this) {\n is Set -> this\n is Collection -> if (this.safeToConvertToSet()) toHashSet() else this\n else -> if (brittleContainsOptimizationEnabled()) toHashSet() else toList()\n }\n\n/**\n * Converts this sequence to a set if [brittleContainsOptimizationEnabled] is true,\n * otherwise converts it to a list.\n *^\ninternal fun <T>
Sequence<T>.convertToSetForSetOperation(): Collection<T> =\n if (brittleContainsOptimizationEnabled()) toHashSet() else toList()\n\n/**\n * Converts this array to a set if [brittleContainsOptimizationEnabled] is true,\n * otherwise converts it to a list.\n *^\ninternal fun <T>
Array<T>.convertToSetForSetOperation(): Collection<T> =\n if (brittleContainsOptimizationEnabled()) toHashSet() else asList()\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\n/**\n * Data class representing a value from a collection or sequence, along with its index in that collection or sequence.\n */\n * @property value the underlying value.\n * @property index the index of the value in the collection or sequence.\n */\n *^\npublic data class IndexedValue<out T>(public val index: Int, public val value: T)\n\n"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmName("MapAccessorsKt")\n\npackage kotlin.collections\n\nimport kotlin.reflect.KProperty\nimport kotlin.internal.Exact\n\n/**\n * Returns the value of the property for the given object from this read-only map.\n * @param thisRef the object for which the value is requested (not used).\n * @param property the metadata for the property, used to get the name of property and lookup the value corresponding to this name in the map.\n * @return the property value.\n * @throws NoSuchElementException when the map doesn't contain value for the property name and doesn't provide an implicit default (see [withDefault]).\n */\n *^\n@kotlin.internal.InlineOnly\npublic inline operator fun <V, V1 : V> Map<in String, @Exact V>.getValue(thisRef: Any?, property: KProperty<*>): V1 =\n @Suppress("UNCHECKED_CAST") (getOrNull(property.name) as V1)\n\n/**\n * Returns the value of the property for the given object from this mutable map.\n * @param thisRef the object for which the value is requested (not used).\n * @param property the metadata for the property, used to get the name of property and lookup the value corresponding to this name in the map.\n * @return the property value.\n * @throws NoSuchElementException when the map doesn't contain value for the property name and doesn't provide an implicit default (see [withDefault]).\n */\n *^\n@kotlin.jvm.JvmName("getVar")\n@kotlin.internal.InlineOnly\npublic inline operator fun <V, V1 : V> MutableMap<in String, out @Exact V>.getValue(thisRef: Any?, property: KProperty<*>): V1 =\n @Suppress("UNCHECKED_CAST") (getOrNull(property.name) as V1)\n\n/**\n * Stores the value of the property for the given object in this mutable map.\n * @param thisRef the object for which the value is requested (not used).\n * @param property the metadata for the property, used to get the name of property and store the value associated with that name in the map.\n * @param value the value to set.\n */\n *^\n@kotlin.internal.InlineOnly\npublic inline operator fun <V> MutableMap<in String, in V>.setValue(thisRef: Any?, property: KProperty<*>, value: V) {\n this.put(property.name, value)\n}\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("MapsKt")\n\npackage kotlin.collections\n\n/**\n * Returns the value for the given key, or the implicit default value for this map.\n * By default no implicit value is provided for maps and a [NoSuchElementException] is thrown.\n * To create a map with implicit default value use [withDefault] method.\n * @throws NoSuchElementException when the map doesn't contain a value for the specified key and no implicit default was provided for that map.\n */\n *^\n@kotlin.jvm.JvmName("getOrNullImplicitDefaultNullable")\n@PublishedApi\ninternal fun <K, V> Map<K, V>.getOrNullImplicitDefault(key: K): V {\n if (this is MapWithDefault)\n return this.getOrNullImplicitDefault(key)\n return getOrNull(key, { throw NoSuchElementException("Key $key

```

```

is missing in the map.})})\n\n/**\n * Returns a wrapper of this read-only map, having the implicit default value
provided with the specified function [defaultValue].\n *\n * This implicit default value is used when the original
map doesn't contain a value for the key specified\n * and a value is obtained with [Map.getValue] function, for
example when properties are delegated to the map.\n *\n * When this map already has an implicit default value
provided with a former call to [withDefault], it is being replaced by this call.\n *\n\npublic fun <K, V> Map<K,
V>.withDefault(defaultValue: (key: K) -> V): Map<K, V> =\n when (this) {\n is MapWithDefault ->
this.map.withDefault(defaultValue)\n else -> MapWithDefaultImpl(this, defaultValue)\n }\n\n\n/**\n * Returns
a wrapper of this mutable map, having the implicit default value provided with the specified function
[defaultValue].\n *\n * This implicit default value is used when the original map doesn't contain a value for the key
specified\n * and a value is obtained with [Map.getValue] function, for example when properties are delegated to the
map.\n *\n * When this map already has an implicit default value provided with a former call to [withDefault], it is
being replaced by this call.\n *\n\n@kotlin.jvm.JvmName("withDefaultMutable")\npublic fun <K, V>
MutableMap<K, V>.withDefault(defaultValue: (key: K) -> V): MutableMap<K, V> =\n when (this) {\n is
MutableMapWithDefault -> this.map.withDefault(defaultValue)\n else -> MutableMapWithDefaultImpl(this,
defaultValue)\n }\n\n\nprivate interface MapWithDefault<K, out V> : Map<K, V> {\n public val map: Map<K,
V>\n public fun getOrImplicitDefault(key: K): V\n}\n\nprivate interface MutableMapWithDefault<K, V> :
MutableMap<K, V>, MapWithDefault<K, V> {\n public override val map: MutableMap<K, V>\n}\n\n\nprivate
class MapWithDefaultImpl<K, out V>(public override val map: Map<K, V>, private val default: (key: K) -> V) :
MapWithDefault<K, V> {\n override fun equals(other: Any?): Boolean = map.equals(other)\n override fun
hashCode(): Int = map.hashCode()\n override fun toString(): String = map.toString()\n override val size: Int get()
= map.size\n override fun isEmpty(): Boolean = map.isEmpty()\n override fun containsKey(key: K): Boolean =
map.containsKey(key)\n override fun containsValue(value: @UnsafeVariance V): Boolean =
map.containsValue(value)\n override fun get(key: K): V? = map.get(key)\n override val keys: Set<K> get() =
map.keys\n override val values: Collection<V> get() = map.values\n override val entries: Set<Map.Entry<K,
V>> get() = map.entries\n override fun getOrImplicitDefault(key: K): V = map.getOrElseNullable(key, {
default(key) })\n}\n\nprivate class MutableMapWithDefaultImpl<K, V>(public override val map: MutableMap<K,
V>, private val default: (key: K) -> V) : MutableMapWithDefault<K, V> {\n override fun equals(other: Any?):
Boolean = map.equals(other)\n override fun hashCode(): Int = map.hashCode()\n override fun toString(): String
= map.toString()\n override val size: Int get() = map.size\n override fun isEmpty(): Boolean = map.isEmpty()\n
override fun containsKey(key: K): Boolean = map.containsKey(key)\n override fun containsValue(value:
@UnsafeVariance V): Boolean = map.containsValue(value)\n override fun get(key: K): V? = map.get(key)\n
override val keys: MutableSet<K> get() = map.keys\n override val values: MutableCollection<V> get() =
map.values\n override val entries: MutableSet<MutableMap.MutableEntry<K, V>> get() = map.entries\n\n
override fun put(key: K, value: V): V? = map.put(key, value)\n override fun remove(key: K): V? =
map.remove(key)\n override fun putAll(from: Map<out K, V>) = map.putAll(from)\n override fun clear() =
map.clear()\n}\n\n\noverride fun getOrImplicitDefault(key: K): V = map.getOrElseNullable(key, { default(key)
})\n}\n\n", "*/\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("CollectionsKt")\n\npackage
kotlin.collections\n\nimport kotlin.random.Random\n\n\n/**\n * Removes a single instance of the specified element
from this\n * collection, if it is present.\n *\n * Allows to overcome type-safety restriction of `remove` that requires
to pass an element of type `E`.\n *\n * @return `true` if the element has been successfully removed; `false` if it was
not present in the collection.\n *\n\n@kotlin.internal.InlineOnly\npublic inline fun <@kotlin.internal.OnlyInputTypes
T> MutableCollection<out T>.remove(element: T): Boolean =\n @Suppress("UNCHECKED_CAST") (this as
MutableCollection<T>).remove(element)\n\n\n/**\n * Removes all of this collection's elements that are also
contained in the specified collection.\n *\n * Allows to overcome type-safety restriction of `removeAll` that requires
to pass a collection of type `Collection<E>`.\n *\n * @return `true` if any of the specified elements was removed
from the collection, `false` if the collection was not modified.\n *\n\n@kotlin.internal.InlineOnly\npublic inline fun

```



```

<@kotlin.internal.OnlyInputTypes T> MutableCollection<out T>.removeAll(elements: Collection<T>): Boolean
= \n @Suppress("UNCHECKED_CAST") (this as MutableCollection<T>).removeAll(elements)\n\n/**\n *
Retains only the elements in this collection that are contained in the specified collection.\n *\n * Allows to
overcome type-safety restriction of `retainAll` that requires to pass a collection of type `Collection<E>`.\n *\n *
@return `true` if any element was removed from the collection, `false` if the collection was not modified.\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun <@kotlin.internal.OnlyInputTypes T> MutableCollection<out
T>.retainAll(elements: Collection<T>): Boolean = \n @Suppress("UNCHECKED_CAST") (this as
MutableCollection<T>).retainAll(elements)\n\n/**\n * Adds the specified [element] to this mutable collection.\n
*/\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> MutableCollection<in T>.plusAssign(element: T)
{\n this.add(element)\n}\n\n/**\n * Adds all elements of the given [elements] collection to this mutable
collection.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> MutableCollection<in
T>.plusAssign(elements: Iterable<T>) {\n this.addAll(elements)\n}\n\n/**\n * Adds all elements of the given
[elements] array to this mutable collection.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T>
MutableCollection<in T>.plusAssign(elements: Array<T>) {\n this.addAll(elements)\n}\n\n/**\n * Adds all
elements of the given [elements] sequence to this mutable collection.\n */\n@kotlin.internal.InlineOnly\npublic
inline operator fun <T> MutableCollection<in T>.plusAssign(elements: Sequence<T>) {\n
this.addAll(elements)\n}\n\n/**\n * Removes a single instance of the specified [element] from this mutable
collection.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> MutableCollection<in
T>.minusAssign(element: T) {\n this.remove(element)\n}\n\n/**\n * Removes all elements contained in the given
[elements] collection from this mutable collection.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun
<T> MutableCollection<in T>.minusAssign(elements: Iterable<T>) {\n this.removeAll(elements)\n}\n\n/**\n *
Removes all elements contained in the given [elements] array from this mutable collection.\n
*/\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> MutableCollection<in T>.minusAssign(elements:
Array<T>) {\n this.removeAll(elements)\n}\n\n/**\n * Removes all elements contained in the given [elements]
sequence from this mutable collection.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T>
MutableCollection<in T>.minusAssign(elements: Sequence<T>) {\n this.removeAll(elements)\n}\n\n/**\n * Adds
all elements of the given [elements] collection to this [MutableCollection].\n */\npublic fun <T>
MutableCollection<in T>.addAll(elements: Iterable<T>): Boolean {\n when (elements) {\n is Collection ->
return addAll(elements)\n else -> {\n var result: Boolean = false\n for (item in elements)\n
if (add(item)) result = true\n return result\n }\n }\n}\n\n/**\n * Adds all elements of the given
[elements] sequence to this [MutableCollection].\n */\npublic fun <T> MutableCollection<in T>.addAll(elements:
Sequence<T>): Boolean {\n var result: Boolean = false\n for (item in elements) {\n if (add(item)) result =
true\n }\n return result\n}\n\n/**\n * Adds all elements of the given [elements] array to this
[MutableCollection].\n */\npublic fun <T> MutableCollection<in T>.addAll(elements: Array<out T>): Boolean {\n
return addAll(elements.asList())\n}\n\n/**\n * Removes all elements from this [MutableCollection] that are also
contained in the given [elements] collection.\n */\npublic fun <T> MutableCollection<in T>.removeAll(elements:
Iterable<T>): Boolean {\n return removeAll(elements.convertToSetForSetOperationWith(this))\n}\n\n/**\n *
Removes all elements from this [MutableCollection] that are also contained in the given [elements] sequence.\n
*/\npublic fun <T> MutableCollection<in T>.removeAll(elements: Sequence<T>): Boolean {\n val set =
elements.convertToSetForSetOperation()\n return set.isNotEmpty() && removeAll(set)\n}\n\n/**\n * Removes all
elements from this [MutableCollection] that are also contained in the given [elements] array.\n */\npublic fun <T>
MutableCollection<in T>.removeAll(elements: Array<out T>): Boolean {\n return elements.isNotEmpty() &&
removeAll(elements.convertToSetForSetOperation())\n}\n\n/**\n * Retains only elements of this
[MutableCollection] that are contained in the given [elements] collection.\n */\npublic fun <T>
MutableCollection<in T>.retainAll(elements: Iterable<T>): Boolean {\n return
retainAll(elements.convertToSetForSetOperationWith(this))\n}\n\n/**\n * Retains only elements of this
[MutableCollection] that are contained in the given [elements] array.\n */\npublic fun <T> MutableCollection<in
T>.retainAll(elements: Array<out T>): Boolean {\n if (elements.isNotEmpty())\n return

```

```

retainAll(elements.convertToSetForSetOperation())\n else\n return retainNothing()\n}\n\n/**\n * Retains only
elements of this [MutableCollection] that are contained in the given [elements] sequence.\n */\npublic fun <T>
MutableCollection<in T>.retainAll(elements: Sequence<T>): Boolean {\n val set =
elements.convertToSetForSetOperation()\n if (set.isNotEmpty())\n return retainAll(set)\n else\n return
retainNothing()\n}\n\nprivate fun MutableCollection<*>.retainNothing(): Boolean {\n val result = isEmpty()\n clear()\n return result\n}\n\n/**\n * Removes all elements from this [MutableIterable] that match the given
[predicate].\n */\n * @return `true` if any element was removed from this collection, or `false` when no elements
were removed and collection was not modified.\n */\npublic fun <T> MutableIterable<T>.removeAll(predicate: (T
-> Boolean): Boolean = filterInPlace(predicate, true)\n\n/**\n * Retains only elements of this [MutableIterable] that
match the given [predicate].\n */\n * @return `true` if any element was removed from this collection, or `false` when
all elements were retained and collection was not modified.\n */\npublic fun <T>
MutableIterable<T>.retainAll(predicate: (T) -> Boolean): Boolean = filterInPlace(predicate, false)\n\nprivate fun
<T> MutableIterable<T>.filterInPlace(predicate: (T) -> Boolean, predicateResultToRemove: Boolean): Boolean {\n
 var result = false\n with(iterator()) {\n while (hasNext())\n if (predicate(next()) ==
predicateResultToRemove) {\n remove()\n result = true\n }\n }\n return
result\n}\n\n/**\n * Removes the element at the specified [index] from this list.\n */\n * In Kotlin one should use the
[MutableList.removeAt] function instead.\n */\n@Deprecated("Use removeAt(index) instead.",
ReplaceWith("removeAt(index)"), level = DeprecationLevel.ERROR)\n\nkotlin.internal.InlineOnly\n\npublic inline
fun <T> MutableList<T>.remove(index: Int): T = removeAt(index)\n\n/**\n * Removes the first element from this
mutable list and returns that removed element, or throws [NoSuchElementException] if this list is empty.\n */\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n\npublic fun <T>
MutableList<T>.removeFirst(): T = if (isEmpty()) throw NoSuchElementException("List is empty.") else
removeAt(0)\n\n/**\n * Removes the first element from this mutable list and returns that removed element, or
returns `null` if this list is empty.\n */\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n\npublic fun <T>
MutableList<T>.removeFirstOrNull(): T? = if (isEmpty()) null else removeAt(0)\n\n/**\n * Removes the last
element from this mutable list and returns that removed element, or throws [NoSuchElementException] if this list is
empty.\n */\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n\npublic fun <T>
MutableList<T>.removeLast(): T = if (isEmpty()) throw NoSuchElementException("List is empty.") else
removeAt(lastIndex)\n\n/**\n * Removes the last element from this mutable list and returns that removed element,
or returns `null` if this list is empty.\n */\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n\npublic fun <T>
MutableList<T>.removeLastOrNull(): T? = if (isEmpty()) null else removeAt(lastIndex)\n\n/**\n * Removes all
elements from this [MutableList] that match the given [predicate].\n */\n * @return `true` if any element was
removed from this collection, or `false` when no elements were removed and collection was not modified.\n */\n
*\npublic fun <T> MutableList<T>.removeAll(predicate: (T) -> Boolean): Boolean = filterInPlace(predicate,
true)\n\n/**\n * Retains only elements of this [MutableList] that match the given [predicate].\n */\n * @return `true`
if any element was removed from this collection, or `false` when all elements were retained and collection was not
modified.\n */\n*\npublic fun <T> MutableList<T>.retainAll(predicate: (T) -> Boolean): Boolean =
filterInPlace(predicate, false)\n\nprivate fun <T> MutableList<T>.filterInPlace(predicate: (T) -> Boolean,
predicateResultToRemove: Boolean): Boolean {\n if (this !is RandomAccess)\n return (this as
MutableIterable<T>).filterInPlace(predicate, predicateResultToRemove)\n var writeIndex: Int = 0\n for
(readIndex in 0..lastIndex) {\n val element = this[readIndex]\n if (predicate(element) ==
predicateResultToRemove)\n continue\n if (writeIndex != readIndex)\n this[writeIndex] =
element\n writeIndex++\n }\n if (writeIndex < size) {\n for (removeIndex in lastIndex downTo
writeIndex)\n removeAt(removeIndex)\n return true\n } else {\n return false\n }\n}\n\n","/*\n *
Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file.

```

```

DO NOT EDIT!\n\npackage kotlin.collections\n\n/** An iterator over a sequence of values of type `Byte`.\n\n*\n\npublic abstract class ByteIterator : Iterator<Byte> {\n override final fun next() = nextByte()\n\n /** Returns the next value in the sequence without boxing. *\n\n public abstract fun nextByte(): Byte\n\n}\n\n/** An iterator over a sequence of values of type `Char`.\n\n*\n\npublic abstract class CharIterator : Iterator<Char> {\n override final fun next() = nextChar()\n\n /** Returns the next value in the sequence without boxing. *\n\n public abstract fun nextChar(): Char\n\n}\n\n/** An iterator over a sequence of values of type `Short`.\n\n*\n\npublic abstract class ShortIterator : Iterator<Short> {\n override final fun next() = nextShort()\n\n /** Returns the next value in the sequence without boxing. *\n\n public abstract fun nextShort(): Short\n\n}\n\n/** An iterator over a sequence of values of type `Int`.\n\n*\n\npublic abstract class IntIterator : Iterator<Int> {\n override final fun next() = nextInt()\n\n /** Returns the next value in the sequence without boxing. *\n\n public abstract fun nextInt(): Int\n\n}\n\n/** An iterator over a sequence of values of type `Long`.\n\n*\n\npublic abstract class LongIterator : Iterator<Long> {\n override final fun next() = nextLong()\n\n /** Returns the next value in the sequence without boxing. *\n\n public abstract fun nextLong(): Long\n\n}\n\n/** An iterator over a sequence of values of type `Float`.\n\n*\n\npublic abstract class FloatIterator : Iterator<Float> {\n override final fun next() = nextFloat()\n\n /** Returns the next value in the sequence without boxing. *\n\n public abstract fun nextFloat(): Float\n\n}\n\n/** An iterator over a sequence of values of type `Double`.\n\n*\n\npublic abstract class DoubleIterator : Iterator<Double> {\n override final fun next() = nextDouble()\n\n /** Returns the next value in the sequence without boxing. *\n\n public abstract fun nextDouble(): Double\n\n}\n\n/** An iterator over a sequence of values of type `Boolean`.\n\n*\n\npublic abstract class BooleanIterator : Iterator<Boolean> {\n override final fun next() = nextBoolean()\n\n /** Returns the next value in the sequence without boxing. *\n\n public abstract fun nextBoolean(): Boolean\n\n}\n\n"/*\n\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n\n\n*\n\n\n@file:kotlin.jvm.JvmMultifileClass\n\n@file:kotlin.jvm.JvmName("CollectionsKt")\n\n\npackage kotlin.collections\n\n\nprivate open class ReversedListReadOnly<out T>(private val delegate: List<T>) : AbstractList<T>() {\n override val size: Int get() = delegate.size\n override fun get(index: Int): T = delegate[reverseElementIndex(index)]\n\n}\n\n\nprivate class ReversedList<T>(private val delegate: MutableList<T>) : AbstractMutableList<T>() {\n override val size: Int get() = delegate.size\n override fun get(index: Int): T = delegate[reverseElementIndex(index)]\n override fun clear() = delegate.clear()\n override fun removeAt(index: Int): T = delegate.removeAt(reverseElementIndex(index))\n override fun set(index: Int, element: T): T = delegate.set(reverseElementIndex(index), element)\n override fun add(index: Int, element: T) {\n delegate.add(reversePositionIndex(index), element)\n }\n\n}\n\n\nprivate fun List<*>.reverseElementIndex(index: Int) =\n if (index in 0..lastIndex) lastIndex - index else throw IndexOutOfBoundsException("Element index $index must be in range [0..lastIndex].")\n\n\nprivate fun List<*>.reversePositionIndex(index: Int) =\n if (index in 0..size) size - index else throw IndexOutOfBoundsException("Position index $index must be in range [0..size].")\n\n\n*/\n\n\n * Returns a reversed read-only view of the original List.\n\n * All changes made in the original list will be reflected in the reversed one.\n\n * @sample samples.collections.ReversedViews.asReversedList\n\n\n*\n\n\npublic fun <T> List<T>.asReversed(): List<T> = ReversedListReadOnly(this)\n\n\n*/\n\n\n * Returns a reversed mutable view of the original mutable List.\n\n * All changes made in the original list will be reflected in the reversed one and vice versa.\n\n * @sample samples.collections.ReversedViews.asReversedMutableList\n\n\n*\n\n\n@kotlin.jvm.JvmName("asReversedMutable")\n\n\npublic fun <T> MutableList<T>.asReversed(): MutableList<T> = ReversedList(this)\n\n\n"/*\n\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n\n\n\n*\n\n\n@file:kotlin.jvm.JvmMultifileClass\n\n@file:kotlin.jvm.JvmName("SequencesKt")\n\n@file:OptIn(ExperimentalTypeInference::class)\n\n\npackage kotlin.sequences\n\n\nimport kotlin.coroutines.*\n\nimport kotlin.coroutines.intrinsics.*\n\nimport kotlin.experimental.ExperimentalTypeInference\n\n\n*/\n\n\n * Builds a [Sequence] lazily yielding values one by one.\n\n * @see kotlin.sequences.generateSequence\n\n * @sample samples.collections.Sequences.Building.buildSequenceYieldAll\n\n * @sample

```

```

samples.collections.Sequences.Building.buildFibonacciSequence\n
*\n@SinceKotlin("1.3")\n@Suppress("DEPRECATION")\npublic fun <T> sequence(@BuilderInference block:
suspend SequenceScope<T>().() -> Unit): Sequence<T> = Sequence { iterator(block) }\n\n/**\n * Builds an [Iterator]
lazily yielding values one by one.\n *\n * @sample samples.collections.Sequences.Building.buildIterator\n *
@sample samples.collections.Iterables.Building.iterable\n
*\n@SinceKotlin("1.3")\n@Suppress("DEPRECATION")\npublic fun <T> iterator(@BuilderInference block:
suspend SequenceScope<T>().() -> Unit): Iterator<T> {\n val iterator = SequenceBuilderIterator<T>()\n
iterator.nextStep = block.createCoroutineUnintercepted(receiver = iterator, completion = iterator)\n return
iterator\n}\n\n/**\n * The scope for yielding values of a [Sequence] or an [Iterator], provides [yield] and [yieldAll]
suspension functions.\n *\n * @see sequence\n * @see iterator\n *\n * @sample
samples.collections.Sequences.Building.buildSequenceYieldAll\n * @sample
samples.collections.Sequences.Building.buildFibonacciSequence\n
*\n@RestrictsSuspension\n@SinceKotlin("1.3")\npublic abstract class SequenceScope<in T> internal
constructor() {\n /**\n * Yields a value to the [Iterator] being built and suspends\n * until the next value is
requested.\n *\n * @sample samples.collections.Sequences.Building.buildSequenceYieldAll\n * @sample
samples.collections.Sequences.Building.buildFibonacciSequence\n */\n public abstract suspend fun yield(value:
T)\n\n /**\n * Yields all values from the `iterator` to the [Iterator] being built\n * and suspends until all these
values are iterated and the next one is requested.\n *\n * The sequence of values returned by the given iterator
can be potentially infinite.\n *\n * @sample samples.collections.Sequences.Building.buildSequenceYieldAll\n
*/\n public abstract suspend fun yieldAll(iterator: Iterator<T>)\n\n /**\n * Yields a collections of values to
the [Iterator] being built\n * and suspends until all these values are iterated and the next one is requested.\n *\n
*/\n * @sample samples.collections.Sequences.Building.buildSequenceYieldAll\n */\n public suspend fun
yieldAll(elements: Iterable<T>) {\n if (elements is Collection && elements.isEmpty()) return\n return
yieldAll(elements.iterator())\n }\n\n /**\n * Yields potentially infinite sequence of values to the [Iterator]
being built\n * and suspends until all these values are iterated and the next one is requested.\n *\n * The
sequence can be potentially infinite.\n *\n * @sample
samples.collections.Sequences.Building.buildSequenceYieldAll\n */\n public suspend fun yieldAll(sequence:
Sequence<T>) = yieldAll(sequence.iterator())\n}\n\nprivate typealias State = Int\nprivate const val
State_NotReady: State = 0\nprivate const val State_ManyNotReady: State = 1\nprivate const val State_ManyReady:
State = 2\nprivate const val State_Ready: State = 3\nprivate const val State_Done: State = 4\nprivate const val
State_Failed: State = 5\nprivate class SequenceBuilderIterator<T> : SequenceScope<T>(), Iterator<T>,
Continuation<Unit> {\n private var state = State_NotReady\n private var nextValue: T? = null\n private var
nextIterator: Iterator<T>? = null\n var nextStep: Continuation<Unit>? = null\n\n override fun hasNext():
Boolean {\n while (true) {\n when (state) {\n State_NotReady -> {}\n
State_ManyNotReady -> {\n if (nextIterator!!.hasNext()) {\n state = State_ManyReady\n
return true\n } else {\n nextIterator = null\n }\n
State_Done -> return false\n State_Ready, State_ManyReady -> return true\n else -> throw
exceptionalState()\n }\n state = State_Failed\n val step = nextStep!!\n nextStep = null\n
step.resume(Unit)\n }\n }\n\n override fun next(): T {\n when (state) {\n State_NotReady,
State_ManyNotReady -> return nextNotReady()\n State_ManyReady -> {\n state =
State_ManyNotReady\n return nextIterator!!.next()\n }\n State_Ready -> {\n state =
State_NotReady\n @Suppress("UNCHECKED_CAST")\n val result = nextValue as T\n
nextValue = null\n return result\n }\n else -> throw exceptionalState()\n }\n }\n\n
private fun nextNotReady(): T {\n if (!hasNext()) throw NoSuchElementException() else return next()\n }\n\n
private fun exceptionalState(): Throwable = when (state) {\n State_Done -> NoSuchElementException()\n
State_Failed -> IllegalStateException("Iterator has failed.")\n else -> IllegalStateException("Unexpected state
of the iterator: $state")\n }\n\n override suspend fun yield(value: T) {\n nextValue = value\n state =
State_Ready\n return suspendCoroutineUninterceptedOrReturn { c -> \n nextStep = c\n

```

```

COROUTINE_SUSPENDED\n }\n }\n\n override suspend fun yieldAll(iterator: Iterator<T>) {\n if
(!iterator.hasNext()) return\n nextIterator = iterator\n state = State_ManyReady\n return
suspendCoroutineUninterceptedOrReturn { c ->\n nextStep = c\n COROUTINE_SUSPENDED\n
}\n }\n\n // Completion continuation implementation\n override fun resumeWith(result: Result<Unit>) {\n
result.getOrThrow() // just rethrow exception if it is there\n state = State_Done\n }\n\n override val context:
CoroutineContext\n get() = EmptyCoroutineContext\n}\n\n/*\n * Copyright 2010-2018 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\ninternal fun
checkWindowSizeStep(size: Int, step: Int) {\n require(size > 0 && step > 0) {\n if (size != step)\n \"Both size $size and step $step must be greater than zero.\"\n else\n \"size $size must be greater than
zero.\"\n }\n}\n\ninternal fun <T> Sequence<T>.windowedSequence(size: Int, step: Int, partialWindows: Boolean,
reuseBuffer: Boolean): Sequence<List<T>> {\n checkWindowSizeStep(size, step)\n return Sequence {\n
windowedIterator(iterator(), size, step, partialWindows, reuseBuffer) }\n}\n\ninternal fun <T>
windowedIterator(iterator: Iterator<T>, size: Int, step: Int, partialWindows: Boolean, reuseBuffer: Boolean):
Iterator<List<T>> {\n if (!iterator.hasNext()) return EmptyIterator\n return iterator<List<T>> {\n val
bufferInitialCapacity = size.coerceAtMost(1024)\n val gap = step - size\n if (gap >= 0) {\n var buffer
= ArrayList<T>(bufferInitialCapacity)\n var skip = 0\n for (e in iterator) {\n if (skip > 0) {\n
skip -= 1; continue }\n buffer.add(e)\n if (buffer.size == size) {\n yield(buffer)\n
 if (reuseBuffer) buffer.clear() else buffer = ArrayList(size)\n skip = gap\n }\n }\n if (buffer.isNotEmpty()) {\n if (partialWindows || buffer.size == size) yield(buffer)\n }\n }\n else {\n var buffer = RingBuffer<T>(bufferInitialCapacity)\n for (e in iterator) {\n
buffer.add(e)\n if (buffer.isFull()) {\n if (buffer.size < size) {\n buffer =
buffer.expanded(maxCapacity = size); continue }\n yield(if (reuseBuffer) buffer else
ArrayList(buffer))\n buffer.removeFirst(step)\n }\n }\n if (partialWindows) {\n
while (buffer.size > step) {\n yield(if (reuseBuffer) buffer else ArrayList(buffer))\n buffer.removeFirst(step)\n }\n if (buffer.isNotEmpty()) yield(buffer)\n }\n }\n}\n\ninternal class MovingSubList<out E>(private val list: List<E>) : AbstractList<E>(), RandomAccess {\n
private var fromIndex: Int = 0\n private var _size: Int = 0\n\n fun move(fromIndex: Int, toIndex: Int) {\n
checkRangeIndexes(fromIndex, toIndex, list.size)\n this.fromIndex = fromIndex\n this._size = toIndex -
fromIndex\n }\n\n override fun get(index: Int): E {\n checkElementIndex(index, _size)\n return
list[fromIndex + index]\n }\n\n override val size: Int get() = _size\n}\n\n\n/**\n * Provides ring buffer
implementation.\n * Buffer overflow is not allowed so [add] doesn't overwrite tail but raises an exception.\n
*/\n\nprivate class RingBuffer<T>(private val buffer: Array<Any?>, filledSize: Int) : AbstractList<T>(),
RandomAccess {\n init {\n require(filledSize >= 0) {\n \"ring buffer filled size should not be negative but it is
$filledSize\"\n }\n require(filledSize <= buffer.size) {\n \"ring buffer filled size: $filledSize cannot be larger than
the buffer size: ${buffer.size}\"\n }\n }\n\n constructor(capacity: Int) : this(arrayOfNulls<Any?>(capacity), 0)\n\n private val capacity = buffer.size\n private var startIndex: Int = 0\n\n override var size: Int = filledSize\n\n private set\n\n override fun get(index: Int): T {\n checkElementIndex(index, size)\n @Suppress(\"UNCHECKED_CAST\")\n return buffer[startIndex.forward(index)] as T\n }\n\n fun isFull() =
size == capacity\n\n override fun iterator(): Iterator<T> = object : AbstractIterator<T>() {\n private var count
= size\n private var index = startIndex\n\n override fun computeNext() {\n if (count == 0) {\n
done()\n } else {\n @Suppress(\"UNCHECKED_CAST\")\n setNext(buffer[index] as
T)\n index = index.forward(1)\n count--\n }\n }\n }\n}\n\n\n@Suppress(\"UNCHECKED_CAST\")\n override fun <T> toArray(array: Array<T>): Array<T> {\n val
result: Array<T?> =\n if (array.size < this.size) array.copyOf(this.size) else array as Array<T?>\n\n val
size = this.size\n\n var widx = 0\n var idx = startIndex\n while (widx < size && idx < capacity) {\n
result[widx] = buffer[idx] as T\n widx++\n idx++\n }\n\n idx = 0\n while (widx <
size) {\n result[widx] = buffer[idx] as T\n widx++\n idx++\n }\n if (result.size >

```

```

this.size) result[this.size] = null\n\n return result as Array<T>\n }\n\n override fun toArray(): Array<Any?>
{\n return toArray(arrayOfNulls(size))\n }\n\n /**\n * Creates a new ring buffer with the capacity equal to
the minimum of [maxCapacity] and 1.5 * [capacity].\n * The returned ring buffer contains the same elements as
this ring buffer.\n */\n fun expanded(maxCapacity: Int): RingBuffer<T> {\n val newCapacity = (capacity +
(capacity shr 1) + 1).coerceAtMost(maxCapacity)\n val newBuffer = if (startIndex == 0)
buffer.copyOf(newCapacity) else toArray(arrayOfNulls(newCapacity))\n return RingBuffer(newBuffer, size)\n }\n\n /**\n * Add [element] to the buffer or fail with [IllegalStateException] if no free space available in the
buffer.\n */\n fun add(element: T) {\n if (isFull()) {\n throw IllegalStateException("ring buffer is
full")\n }\n buffer[startIndex.forward(size)] = element\n size++\n }\n\n /**\n * Removes [n]
first elements from the buffer or fails with [IllegalArgumentException] if not enough elements in the buffer to
remove.\n */\n fun removeFirst(n: Int) {\n require(n >= 0) { "\"n shouldn't be negative but it is $n\""}\n require(n <= size) { "\"n shouldn't be greater than the buffer size: n = $n, size = $size\""}\n if (n > 0) {\n val start = startIndex\n val end = start.forward(n)\n if (start > end) {\n buffer.fill(null, start,
capacity)\n buffer.fill(null, 0, end)\n } else {\n buffer.fill(null, start, end)\n }\n startIndex = end\n size -= n\n }\n }\n\n @Suppress("NOTHING_TO_INLINE")\n private
inline fun Int.forward(n: Int): Int = (this + n) % capacity\n}\n\n"/**\n * Copyright 2010-2019 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\n// UByteArray

```

```

=====\n@Exp
erimentalUnsignedTypes\nprivate fun partition(\n array: UByteArray, left: Int, right: Int): Int {\n var i = left\n var j = right\n val pivot = array[(left + right) / 2]\n while (i <= j) {\n while (array[i] < pivot)\n i++\n while (array[j] > pivot)\n j--\n if (i <= j) {\n val tmp = array[i]\n array[i] = array[j]\n array[j] = tmp\n i++\n j--\n }\n }\n return i\n}\n\n@ExperimentalUnsignedTypes\nprivate fun
quickSort(\n array: UByteArray, left: Int, right: Int) {\n val index = partition(array, left, right)\n if (left < index
- 1)\n quickSort(array, left, index - 1)\n if (index < right)\n quickSort(array, index, right)\n}\n\n//
UShortArray

```

```

=====\n@Exp
erimentalUnsignedTypes\nprivate fun partition(\n array: UShortArray, left: Int, right: Int): Int {\n var i = left\n var j = right\n val pivot = array[(left + right) / 2]\n while (i <= j) {\n while (array[i] < pivot)\n i++\n while (array[j] > pivot)\n j--\n if (i <= j) {\n val tmp = array[i]\n array[i] = array[j]\n array[j] = tmp\n i++\n j--\n }\n }\n return i\n}\n\n@ExperimentalUnsignedTypes\nprivate fun
quickSort(\n array: UShortArray, left: Int, right: Int) {\n val index = partition(array, left, right)\n if (left <
index - 1)\n quickSort(array, left, index - 1)\n if (index < right)\n quickSort(array, index, right)\n}\n\n//
UIntArray

```

```

=====\n@Exp
erimentalUnsignedTypes\nprivate fun partition(\n array: UIntArray, left: Int, right: Int): Int {\n var i = left\n var j = right\n val pivot = array[(left + right) / 2]\n while (i <= j) {\n while (array[i] < pivot)\n i++\n while (array[j] > pivot)\n j--\n if (i <= j) {\n val tmp = array[i]\n array[i] = array[j]\n array[j] = tmp\n i++\n j--\n }\n }\n return i\n}\n\n@ExperimentalUnsignedTypes\nprivate fun
quickSort(\n array: UIntArray, left: Int, right: Int) {\n val index = partition(array, left, right)\n if (left < index
- 1)\n quickSort(array, left, index - 1)\n if (index < right)\n quickSort(array, index, right)\n}\n\n//
ULongArray

```

```

=====\n@Exp
erimentalUnsignedTypes\nprivate fun partition(\n array: ULongArray, left: Int, right: Int): Int {\n var i = left\n var j = right\n val pivot = array[(left + right) / 2]\n while (i <= j) {\n while (array[i] < pivot)\n i++\n while (array[j] > pivot)\n j--\n if (i <= j) {\n val tmp = array[i]\n array[i] = array[j]\n array[j] = tmp\n i++\n j--\n }\n }\n return i\n}\n\n@ExperimentalUnsignedTypes\nprivate fun
quickSort(\n array: ULongArray, left: Int, right: Int) {\n val index = partition(array, left, right)\n if (left < index

```

```
- 1)\n quickSort(array, left, index - 1)\n if (index < right)\n quickSort(array, index, right)\n}\n\n\n//
```

## Interfaces

```
=====\n\n\n * Sorts the given array using qsort algorithm.\n * \n\n @ExperimentalUnsignedTypes\n internal fun sortArray(array: UByteArray, fromIndex: Int, toIndex: Int) = quickSort(array, fromIndex, toIndex - 1)\n\n @ExperimentalUnsignedTypes\n internal fun sortArray(array: UShortArray, fromIndex: Int, toIndex: Int) = quickSort(array, fromIndex, toIndex - 1)\n\n @ExperimentalUnsignedTypes\n internal fun sortArray(array: UIntArray, fromIndex: Int, toIndex: Int) = quickSort(array, fromIndex, toIndex - 1)\n\n @ExperimentalUnsignedTypes\n internal fun sortArray(array: ULongArray, fromIndex: Int, toIndex: Int) = quickSort(array, fromIndex, toIndex - 1), /*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n package kotlin\n\n import kotlin.internal.InlineOnly\n\n /**\n * Compares this object with the specified object for order. Returns zero if this object is equal\n * to the specified [other] object, a negative number if it's less than [other], or a positive number\n * if it's greater than [other].\n * This function delegates to [Comparable.compareTo] and allows to call it in infix form.\n *\n @InlineOnly\n @SinceKotlin("1.6")\n public inline infix fun <T> Comparable<T>.compareTo(other: T): Int =\n this.compareTo(other)\n", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n package kotlin.contracts\n\n import kotlin.internal.ContractsDsl\n\n import kotlin.internal.InlineOnly\n\n /**\n * This marker distinguishes the experimental contract declaration API and is used to opt-in for that feature\n * when declaring contracts of user functions.\n * Any usage of a declaration annotated with `@ExperimentalContracts` must be accepted either by\n * annotating that usage with the [OptIn] annotation, e.g. `@OptIn(ExperimentalContracts::class)`,\n * or by using the compiler argument `-opt-in=kotlin.contracts.ExperimentalContracts`.\n *\n @Retention(AnnotationRetention.BINARY)\n @SinceKotlin("1.3")\n @RequiresOptIn\n @MustBeDocumente\n d\n public annotation class ExperimentalContracts\n\n /**\n * Provides a scope, where the functions of the contract DSL, such as [returns], [callsInPlace], etc.,\n * can be used to describe the contract of a function.\n * This type is used as a receiver type of the lambda function passed to the [contract] function.\n * @see contract\n *\n @ContractsDsl\n @ExperimentalContracts\n @SinceKotlin("1.3")\n public interface ContractBuilder {\n\n /**\n * Describes a situation when a function returns normally, without any exceptions thrown.\n * Use [SimpleEffect.implies] function to describe a conditional effect that happens in such case.\n * \n // @sample samples.contracts.returnsContract\n @ContractsDsl\n public fun returns(): Returns\n\n /**\n * Describes a situation when a function returns normally with the specified return [value].\n * The possible values of [value] are limited to `true`, `false` or `null`.\n * Use [SimpleEffect.implies] function to describe a conditional effect that happens in such case.\n * \n // @sample samples.contracts.returnsTrueContract\n // @sample samples.contracts.returnsFalseContract\n // @sample samples.contracts.returnsNullContract\n @ContractsDsl\n public fun returns(value: Any?): Returns\n\n /**\n * Describes a situation when a function returns normally with any value that is not `null`.\n * Use [SimpleEffect.implies] function to describe a conditional effect that happens in such case.\n * \n // @sample samples.contracts.returnsNotNullContract\n @ContractsDsl\n public fun returnsNotNull(): ReturnsNotNull\n\n /**\n * Specifies that the function parameter [lambda] is invoked in place.\n * This contract specifies that:\n * 1. the function [lambda] can only be invoked during the call of the owner function,\n * and it won't be invoked after that owner function call is completed;\n * 2. _(optionally)_ the function [lambda] is invoked the amount of times specified by the [kind] parameter.\n * see the [InvocationKind] enum for possible values.\n *\n * A function declaring the `callsInPlace` effect must be _inline_.\n * \n // @sample samples.contracts.callsInPlaceAtMostOnceContract\n * @sample samples.contracts.callsInPlaceAtLeastOnceContract\n * @sample samples.contracts.callsInPlaceExactlyOnceContract\n * @sample
```

```

samples.contracts.callsInPlaceUnknownContract\n */\n @ContractsDsl public fun <R> callsInPlace(lambda:
Function<R>, kind: InvocationKind = InvocationKind.UNKNOWN): CallsInPlace\n}\n\n/**\n * Specifies how
many times a function invokes its function parameter in place.\n *\n * See [ContractBuilder.callsInPlace] for the
details of the call-in-place function contract.\n
*/\n\n@ContractsDsl\n@ExperimentalContracts\n@SinceKotlin("1.3")\npublic enum class InvocationKind {\n
/**\n * A function parameter will be invoked one time or not invoked at all.\n */\n // @sample
samples.contracts.callsInPlaceAtMostOnceContract\n @ContractsDsl AT_MOST_ONCE,\n\n /**\n * A
function parameter will be invoked one or more times.\n */\n // @sample
samples.contracts.callsInPlaceAtLeastOnceContract\n @ContractsDsl AT_LEAST_ONCE,\n\n /**\n * A
function parameter will be invoked exactly one time.\n */\n // @sample
samples.contracts.callsInPlaceExactlyOnceContract\n @ContractsDsl EXACTLY_ONCE,\n\n /**\n * A
function parameter is called in place, but it's unknown how many times it can be called.\n */\n // @sample
samples.contracts.callsInPlaceUnknownContract\n @ContractsDsl UNKNOWN\n}\n\n/**\n * Specifies the
contract of a function.\n *\n * The contract description must be at the beginning of a function and have at least one
effect.\n *\n * Only the top-level functions can have a contract for now.\n *\n * @param builder the lambda where
the contract of a function is described with the help of the [ContractBuilder] members.\n *\n */\n\n@sample
samples.contracts.returnsContract\n* @sample samples.contracts.returnsTrueContract\n* @sample
samples.contracts.returnsFalseContract\n* @sample samples.contracts.returnsNullContract\n* @sample
samples.contracts.returnsNotNullContract\n* @sample samples.contracts.callsInPlaceAtMostOnceContract\n*
@sample samples.contracts.callsInPlaceAtLeastOnceContract\n* @sample
samples.contracts.callsInPlaceExactlyOnceContract\n* @sample
samples.contracts.callsInPlaceUnknownContract\n*/\n\n@ContractsDsl\n@ExperimentalContracts\n@InlineOnly\n\n@
SinceKotlin("1.3")\n\n@Suppress("UNUSED_PARAMETER")\n\npublic inline fun contract(builder:
ContractBuilder.() -> Unit) { }\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.coroutines\n\n/**\n * Marks coroutine context element that
intercepts coroutine continuations.\n * The coroutines framework uses [ContinuationInterceptor.Key] to retrieve the
interceptor and\n * intercepts all coroutine continuations with [interceptContinuation] invocations.\n *\n *
[ContinuationInterceptor] behaves like a [polymorphic element][AbstractCoroutineContextKey], meaning that\n *
its implementation delegates [get][CoroutineContext.Element.get] and
[minusKey][CoroutineContext.Element.minusKey]\n * to [getPolymorphicElement] and [minusPolymorphicKey]
respectively.\n * [ContinuationInterceptor] subtypes can be extracted from the coroutine context using either
[ContinuationInterceptor.Key]\n * or subtype key if it extends [AbstractCoroutineContextKey].\n
*/\n\n@SinceKotlin("1.3")\n\npublic interface ContinuationInterceptor : CoroutineContext.Element {\n\n /**\n *
The key that defines *the* context interceptor.\n */\n companion object Key :
CoroutineContext.Key<ContinuationInterceptor>\n\n /**\n * Returns continuation that wraps the original
[continuation], thus intercepting all resumptions.\n */\n * This function is invoked by coroutines framework when
needed and the resulting continuations are\n * cached internally per each instance of the original [continuation].\n
*/\n * This function may simply return original [continuation] if it does not want to intercept this particular
continuation.\n */\n * When the original [continuation] completes, coroutine framework invokes
[releaseInterceptedContinuation]\n * with the resulting continuation if it was intercepted, that is if
`interceptContinuation` had previously\n * returned a different continuation instance.\n */\n public fun <T>
interceptContinuation(continuation: Continuation<T>): Continuation<T>\n\n /**\n * Invoked for the
continuation instance returned by [interceptContinuation] when the original\n * continuation completes and will
not be used anymore. This function is invoked only if [interceptContinuation]\n * had returned a different
continuation instance from the one it was invoked with.\n */\n * Default implementation does nothing.\n */\n
* @param continuation Continuation instance returned by this interceptor's [interceptContinuation] invocation.\n
/\n\n public fun releaseInterceptedContinuation(continuation: Continuation<>) {\n\n /* do nothing by default

```



```

*\/n }\/n\n public override operator fun <E : CoroutineContext.Element> get(key: CoroutineContext.Key<E>):
E? {\/n // getPolymorphicKey specialized for ContinuationInterceptor key\/n
@OptIn(ExperimentalStdlibApi::class)\n if (key is AbstractCoroutineContextKey<*, *>) {\/n
@Suppress("\UNCHECKED_CAST")\n return if (key.isSubKey(this.key)) key.tryCast(this) as? E else
null\/n }\/n @Suppress("\UNCHECKED_CAST")\n return if (ContinuationInterceptor === key) this as
E else null\/n }\/n\n public override fun minusKey(key: CoroutineContext.Key<*>): CoroutineContext {\/n
// minusPolymorphicKey specialized for ContinuationInterceptor key\/n
@OptIn(ExperimentalStdlibApi::class)\n if (key is AbstractCoroutineContextKey<*, *>) {\/n return if
(key.isSubKey(this.key) && key.tryCast(this) != null) EmptyCoroutineContext else this\/n }\/n return if
(ContinuationInterceptor === key) EmptyCoroutineContext else this\/n }\/n\n", "*/n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */n\npackage kotlin.coroutines\n\n/**\n *
Persistent context for the coroutine. It is an indexed set of [Element] instances.\n * An indexed set is a mix between
a set and a map.\n * Every element in this set has a unique [Key].\n */n\n@SinceKotlin("1.3")\npublic interface
CoroutineContext {\/n /**\n * Returns the element with the given [key] from this context or `null`.\/n */n
*/n\n public operator fun <E : Element> get(key: Key<E>): E?\/n /**\n * Accumulates entries of this context
starting with [initial] value and applying [operation]\n * from left to right to current accumulator value and each
element of this context.\n */n\n public fun <R> fold(initial: R, operation: (R, Element) -> R): R\/n /**\n *
Returns a context containing elements from this context and elements from other [context].\n * The elements
from this context with the same key as in the other one are dropped.\n */n\n public operator fun plus(context:
CoroutineContext): CoroutineContext =\/n if (context === EmptyCoroutineContext) this else // fast path -- avoid
lambda creation\n context.fold(this) { acc, element ->\n val removed =
acc.minusKey(element.key)\n if (removed === EmptyCoroutineContext) element else {\/n //
make sure interceptor is always last in the context (and thus is fast to get when present)\n val interceptor
= removed[ContinuationInterceptor]\n if (interceptor == null) CombinedContext(removed, element) else
{\/n val left = removed.minusKey(ContinuationInterceptor)\n if (left ===
EmptyCoroutineContext) CombinedContext(element, interceptor) else\n
CombinedContext(CombinedContext(left, element), interceptor)\n }\/n }\/n }\/n }\/n\n /**\n
* Returns a context containing elements from this context, but without an element with\n * the specified [key].\n
/n\n public fun minusKey(key: Key<>): CoroutineContext\n /**\n * Key for the elements of
[CoroutineContext]. [E] is a type of element with this key.\n */n\n public interface Key<E : Element>\n /**\n
* An element of the [CoroutineContext]. An element of the coroutine context is a singleton context by itself.\n
*/n\n public interface Element : CoroutineContext {\/n /**\n * A key of this coroutine context element.\n
/n\n public val key: Key<>\n\n public override operator fun <E : Element> get(key: Key<E>): E? =\/n
@Suppress("\UNCHECKED_CAST")\n if (this.key == key) this as E else null\/n\n public override
fun <R> fold(initial: R, operation: (R, Element) -> R): R =\/n operation(initial, this)\n\n public override
fun minusKey(key: Key<*>): CoroutineContext =\/n if (this.key == key) EmptyCoroutineContext else this\/n
}\/n }\/n\n", "*/n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/n\n\npackage kotlin.coroutines\n\nimport kotlin.coroutines.CoroutineContext.Element\nimport
kotlin.coroutines.CoroutineContext.Key\n\n/**\n * Base class for [CoroutineContext.Element] implementations.\n
*/n\n@SinceKotlin("1.3")\npublic abstract class AbstractCoroutineContextElement(public override val key:
Key<*>) : Element\n\n/**\n * Base class for [CoroutineContext.Key] associated with polymorphic
[CoroutineContext.Element] implementation.\n * Polymorphic element implementation implies delegating its
[get][Element.get] and [minusKey][Element.minusKey]\n * to [getPolymorphicElement] and
[minusPolymorphicKey] respectively.\n */n\n * Polymorphic elements can be extracted from the coroutine context
using both element key and its supertype key.\n * Example of polymorphic elements:\n * ```\n * open class
BaseElement : CoroutineContext.Element {\/n * companion object Key : CoroutineContext.Key<BaseElement>\n

```

```

* override val key: CoroutineContext.Key<*> get() = Key\n * // It is important to use getPolymorphicKey and
minusPolymorphicKey\n * override fun <E : CoroutineContext.Element> get(key: CoroutineContext.Key<E>):
E? = getPolymorphicElement(key)\n * override fun minusKey(key: CoroutineContext.Key<*>):
CoroutineContext = minusPolymorphicKey(key)\n * }\n * \n * class DerivedElement : BaseElement() {\n *
companion object Key : AbstractCoroutineContextKey<BaseElement, DerivedElement>(BaseElement, { it as?
DerivedElement })\n * }\n * // Now it is possible to query both `BaseElement` and `DerivedElement`\n *
someContext[BaseElement] // Returns BaseElement?, non-null both for BaseElement and DerivedElement
instances\n * someContext[DerivedElement] // Returns DerivedElement?, non-null only for DerivedElement
instance\n * ```\n * @param B base class of a polymorphic element\n * @param baseKey an instance of base key\n
* @param E element type associated with the current key\n * @param safeCast a function that can safely cast
abstract [CoroutineContext.Element] to the concrete [E] type\n * and return the element if it is a subtype
of [E] or `null` otherwise.\n * \n * \n * @SinceKotlin("1.3")\n * @ExperimentalStdlibApi\n * public abstract class
AbstractCoroutineContextKey<B : Element, E : B>(\n * baseKey: Key,\n * private val safeCast: (element:
Element) -> E?)\n *) : Key<E> {\n * private val topmostKey: Key<*> = if (baseKey is
AbstractCoroutineContextKey<*, *>) baseKey.topmostKey else baseKey\n * \n * internal fun tryCast(element:
Element): E? = safeCast(element)\n * internal fun isSubKey(key: Key<*>): Boolean = key === this || topmostKey
=== key\n * }\n * \n * Returns the current element if it is associated with the given [key] in a polymorphic manner
or `null` otherwise.\n * This method returns non-null value if either [Element.key] is equal to the given [key] or if
the [key] is associated\n * with [Element.key] via [AbstractCoroutineContextKey].\n * See
[AbstractCoroutineContextKey] for the example of usage.\n
* \n * @SinceKotlin("1.3")\n * @ExperimentalStdlibApi\n * public fun <E : Element>
Element.getPolymorphicElement(key: Key<E>): E? {\n * if (key is AbstractCoroutineContextKey<*, *>) {\n *
@Suppress("UNCHECKED_CAST")\n * return if (key.isSubKey(this.key)) key.tryCast(this) as? E else null\n *
}\n * @Suppress("UNCHECKED_CAST")\n * return if (this.key === key) this as E else null\n * }\n * \n * Returns empty coroutine context if the element is associated with the given [key] in a polymorphic manner\n * or
`null` otherwise.\n * This method returns empty context if either [Element.key] is equal to the given [key] or if the
[key] is associated\n * with [Element.key] via [AbstractCoroutineContextKey].\n * See
[AbstractCoroutineContextKey] for the example of usage.\n
* \n * @SinceKotlin("1.3")\n * @ExperimentalStdlibApi\n * public fun Element.minusPolymorphicKey(key: Key<*>):
CoroutineContext {\n * if (key is AbstractCoroutineContextKey<*, *>) {\n * return if (key.isSubKey(this.key)
&& key.tryCast(this) != null) EmptyCoroutineContext else this\n * }\n * return if (this.key === key)
EmptyCoroutineContext else this\n * }\n * \n * An empty coroutine context.\n * \n * @SinceKotlin("1.3")\n * public
object EmptyCoroutineContext : CoroutineContext, Serializable {\n * private const val serialVersionUID: Long =
0\n * private fun readResolve(): Any = EmptyCoroutineContext\n * public override fun <E : Element> get(key:
Key<E>): E? = null\n * public override fun <R> fold(initial: R, operation: (R, Element) -> R): R = initial\n *
public override fun plus(context: CoroutineContext): CoroutineContext = context\n * public override fun minusKey(key:
Key<*>): CoroutineContext = this\n * public override fun hashCode(): Int = 0\n * public override fun toString():
String = "EmptyCoroutineContext"\n * }\n * \n * ----- internal impl -----\n * \n * // this class is not
exposed, but is hidden inside implementations\n * // this is a left-biased list, so that `plus` works
naturally\n * \n * @SinceKotlin("1.3")\n * internal class CombinedContext(\n * private val left: CoroutineContext,\n *
private val element: Element\n *) : CoroutineContext, Serializable {\n * override fun <E : Element> get(key:
Key<E>): E? {\n * var cur = this\n * while (true) {\n * cur.element[key]?.let { return it }\n * val next
= cur.left\n * if (next is CombinedContext) {\n * cur = next\n * } else {\n * return
next[key]\n * }\n * }\n * }\n * public override fun <R> fold(initial: R, operation: (R, Element) -> R): R =\n *
operation(left.fold(initial, operation), element)\n * public override fun minusKey(key: Key<*>):
CoroutineContext {\n * element[key]?.let { return left }\n * val newLeft = left.minusKey(key)\n * return
when {\n * newLeft === left -> this\n * newLeft === EmptyCoroutineContext -> element\n * else ->
CombinedContext(newLeft, element)\n * }\n * }\n * private fun size(): Int {\n * var cur = this\n * var size

```

```

= 2\n while (true) {\n cur = cur.left as? CombinedContext ?: return size\n size++\n }\n }\n\n private fun contains(element: Element): Boolean =\n get(element.key) == element\n\n private fun\ncontainsAll(context: CombinedContext): Boolean {\n var cur = context\n while (true) {\n if\n(!contains(cur.element)) return false\n val next = cur.left\n if (next is CombinedContext) {\n cur = next\n } else {\n return contains(next as Element)\n }\n }\n }\n\n override fun\nequals(other: Any?): Boolean =\n this === other || other is CombinedContext && other.size() == size() &&\n other.containsAll(this)\n\n override fun hashCode(): Int = left.hashCode() + element.hashCode()\n\n override\nfun toString(): String =\n "[" + fold("") { acc, element ->\n if (acc.isEmpty()) element.toString() else\n"$acc, $element"\n } + "]\n\n private fun writeReplace(): Any {\n val n = size()\n val elements =\n arrayOfNulls<CoroutineContext>(n)\n var index = 0\n fold(Unit) { _, element -> elements[index++] =\n element }\n check(index == n)\n @Suppress("UNCHECKED_CAST")\n return Serialized(elements\n as Array<CoroutineContext>)\n }\n\n private class Serialized(val elements: Array<CoroutineContext>) :\n Serializable {\n companion object {\n private const val serialVersionUID: Long = 0L\n }\n\n private fun readResolve(): Any = elements.fold(EmptyCoroutineContext, CoroutineContext::plus)\n }\n}\n"/\n\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code\nis governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n\n*/\n\n@file:kotlin.jvm.JvmName("IntrinsicsKt")\n@file:kotlin.jvm.JvmMultifileClass\n\npackage\nkotlin.coroutines.intrinsics\n\nimport kotlin.contracts.*\nimport kotlin.coroutines.*\nimport\nkotlin.internal.InlineOnly\n\n/**\n * Obtains the current continuation instance inside suspend functions and either\nsuspends\n * currently running coroutine or returns result immediately without suspension.\n * If the [block]\nreturns the special [COROUTINE_SUSPENDED] value, it means that suspend function did suspend the execution\nand will\n * not return any result immediately. In this case, the [Continuation] provided to the [block] shall be\n * resumed by invoking [Continuation.resumeWith] at some moment in the\n * future when the result becomes\navailable to resume the computation.\n * Otherwise, the return value of the [block] must have a type assignable\nto [T] and represents the result of this suspend function.\n * It means that the execution was not suspended and the\n[Continuation] provided to the [block] shall not be invoked.\n * As the result type of the [block] is declared as\n`Any?` and cannot be correctly type-checked,\n * its proper return type remains on the conscience of the suspend\nfunction's author.\n * Invocation of [Continuation.resumeWith] resumes coroutine directly in the invoker's\nthread without going through the\n * [ContinuationInterceptor] that might be present in the coroutine's\n[CoroutineContext].\n * It is the invoker's responsibility to ensure that a proper invocation context is established.\n * [Continuation.intercepted] can be used to acquire the intercepted continuation.\n * Note that it is not\nrecommended to call either [Continuation.resume] nor [Continuation.resumeWithException] functions\nsynchronously\n * in the same stackframe where suspension function is run. Use [suspendCoroutine] as a safer way\nto obtain current\n * continuation instance.\n\n*/\n\n@SinceKotlin("1.3")\n@InlineOnly\n@Suppress("UNUSED_PARAMETER",\n"RedundantSuspendModifier")\npublic suspend inline fun <T>\nsuspendCoroutineUninterceptedOrReturn(crossinline block: (Continuation<T>) -> Any?): T {\n contract {\n callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n }\n throw NotImplementedError("Implementation of\nsuspendCoroutineUninterceptedOrReturn is intrinsic")\n}\n\n/**\n * This value is used as a return value of\n[suspendCoroutineUninterceptedOrReturn] `block` argument to state that\n * the execution was suspended and will\nnot return any result immediately.\n * **Note: this value should not be used in general code.** Using it outside\nof the context of\n * `suspendCoroutineUninterceptedOrReturn` function return value (including, but not limited\nto,\n * storing this value in other properties, returning it from other functions, etc)\n * can lead to unspecified\nbehavior of the code.\n */\n\n// It is implemented as property with getter to avoid ProGuard <clinit> problem with\nmultifile IntrinsicsKt class\n@SinceKotlin("1.3")\npublic val COROUTINE_SUSPENDED: Any get() =\n CoroutineSingletons.COROUTINE_SUSPENDED\n\n// Using enum here ensures two important properties:\n// 1.\nIt makes SafeContinuation serializable with all kinds of serialization frameworks (since all of them natively support\nenums)\n// 2. It improves debugging experience, since you clearly see toString() value of those objects and what

```

package they come from\n@SinceKotlin("1.3")\n@PublishedApi // This class is Published API via serialized representation of SafeContinuation, don't rename/move\ninternal enum class CoroutineSingletons { COROUTINE\_SUSPENDED, UNDECIDED, RESUMED }\n", "/\*\n \* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n \* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n \*/\npackage kotlin.experimental\n\n/\*\* Performs a bitwise AND operation between the two values. \*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Byte.and(other: Byte): Byte = (this.toInt() and other.toInt()).toByte()\n\n/\*\* Performs a bitwise OR operation between the two values. \*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Byte.or(other: Byte): Byte = (this.toInt() or other.toInt()).toByte()\n\n/\*\* Performs a bitwise XOR operation between the two values. \*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Byte.xor(other: Byte): Byte = (this.toInt() xor other.toInt()).toByte()\n\n/\*\* Inverts the bits in this value. \*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun Byte.inv(): Byte = (this.toInt().inv()).toByte()\n\n/\*\* Performs a bitwise AND operation between the two values. \*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Short.and(other: Short): Short = (this.toInt() and other.toInt()).toShort()\n\n/\*\* Performs a bitwise OR operation between the two values. \*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Short.or(other: Short): Short = (this.toInt() or other.toInt()).toShort()\n\n/\*\* Performs a bitwise XOR operation between the two values. \*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Short.xor(other: Short): Short = (this.toInt() xor other.toInt()).toShort()\n\n/\*\* Inverts the bits in this value. \*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun Short.inv(): Short = (this.toInt().inv()).toShort()\n\n", "/\*\n \* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n \* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n \*/\npackage kotlin.experimental\n\n/\*\*\n \* The experimental marker for type inference augmenting annotations.\n \* Any usage of a declaration annotated with `@ExperimentalTypeInference` must be accepted either by\n \* annotating that usage with the [OptIn] annotation, e.g. `@OptIn(ExperimentalTypeInference::class)`,\n \* or by using the compiler argument `-opt-in=kotlin.experimental.ExperimentalTypeInference`.\n \*/\n@RequiresOptIn(level = RequiresOptIn.Level.ERROR)\n@MustBeDocumented\n@Retention(AnnotationRetention.BINARY)\n@Target(AnnotationTarget.ANNOTATION\_CLASS)\n@SinceKotlin("1.3")\npublic annotation class ExperimentalTypeInference\n", "/\*\n \* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n \* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n \*/\npackage kotlin.internal\n\n/\*\*\n \* Specifies that the corresponding type should be ignored during type inference.\n \*/\n@Target(AnnotationTarget.TYPE)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class NoInfer\n\n/\*\*\n \* Specifies that the constraint built for the type during type inference should be an equality one.\n \*/\n@Target(AnnotationTarget.TYPE)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class Exact\n\n/\*\*\n \* Specifies that a corresponding member has the lowest priority in overload resolution.\n \*/\n@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY, AnnotationTarget.CONSTRUCTOR)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class LowPriorityInOverloadResolution\n\n/\*\*\n \* Specifies that the corresponding member has the highest priority in overload resolution. Effectively this means that\n \* an extension annotated with this annotation will win in overload resolution over a member with the same signature.\n \*/\n@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class HidesMembers\n\n/\*\*\n \* The value of this type parameter should be mentioned in input types (argument types, receiver type or expected type).\n \*/\n@Target(AnnotationTarget.TYPE\_PARAMETER)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class OnlyInputTypes\n\n/\*\*\n \* Specifies that this function should not be called directly without inlining\n \*/\n@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY,

AnnotationTarget.PROPERTY\_GETTER,  
AnnotationTarget.PROPERTY\_SETTER)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class  
InlineOnly\n\n/\*\n \* Specifies that this declaration can have dynamic receiver type.\n  
\*/\n@Target(AnnotationTarget.FUNCTION,  
AnnotationTarget.PROPERTY)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class  
DynamicExtension\n\n/\*\n \* The value of this parameter should be a property reference expression (`this::foo`),  
referencing a `lateinit` property,\n \* the backing field of which is accessible at the point where the corresponding  
argument is passed.\n  
\*/\n@Target(AnnotationTarget.VALUE\_PARAMETER)\n@Retention(AnnotationRetention.BINARY)\n@SinceK  
otlin("1.2")\ninternal annotation class AccessibleLateinitPropertyLiteral\n\n/\*\n \* Specifies that this declaration is  
only completely supported since the specified version.\n \* The Kotlin compiler of an earlier version is going to  
report a diagnostic on usages of this declaration.\n \* The diagnostic message can be specified with [message], or via  
[errorCode] (takes less space, but might not be immediately clear\n \* to the user). The diagnostic severity can be  
specified with [level]: WARNING/ERROR mean that either a warning or an error\n \* is going to be reported,  
HIDDEN means that the declaration is going to be removed from resolution completely.\n \* [versionKind]  
specifies which version should be compared with the [version] value, when compiling the usage of the annotated  
declaration.\n \* Note that prior to 1.2, only [RequireKotlinVersionKind.LANGUAGE\_VERSION] was supported,  
so the Kotlin compiler before 1.2 is going to\n \* treat any [RequireKotlin] as if it requires the language version.  
Since 1.2, the Kotlin compiler supports\n \* [RequireKotlinVersionKind.LANGUAGE\_VERSION],  
[RequireKotlinVersionKind.COMPILER\_VERSION] and [RequireKotlinVersionKind.API\_VERSION].\n \* If the  
actual value of [versionKind] is something different (e.g. a new version kind, added in future versions of Kotlin),\n \*  
Kotlin 1.2 is going to ignore this [RequireKotlin] altogether, where as Kotlin before 1.2 is going to treat this as a  
requirement\n \* on the language version.\n \* This annotation is erased at compile time; its arguments are stored  
in a more compact form in the Kotlin metadata.\n \* \n@Target(AnnotationTarget.CLASS,  
AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY, AnnotationTarget.CONSTRUCTOR,  
AnnotationTarget.TYPEALIAS)\n@Retention(AnnotationRetention.SOURCE)\n@Repeatable\n@SinceKotlin("1.  
2")\ninternal annotation class RequireKotlin(\n val version: String,\n val message: String = "",\n val level:  
DeprecationLevel = DeprecationLevel.ERROR,\n val versionKind: RequireKotlinVersionKind =  
RequireKotlinVersionKind.LANGUAGE\_VERSION,\n val errorCode: Int = -1)\n\n/\*\n \* The kind of the  
version that is required by [RequireKotlin].\n \* \n@SinceKotlin("1.2")\ninternal enum class  
RequireKotlinVersionKind {\n LANGUAGE\_VERSION,\n COMPILER\_VERSION,\n  
API\_VERSION,\n}\n\n/\*\n \* Specifies that this declaration is a part of special DSL, used for constructing  
function's contract.\n \* \n@Retention(AnnotationRetention.BINARY)\n@SinceKotlin("1.2")\ninternal annotation  
class ContractsDsl\n\n/\*\n \* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language  
contributors.\n \* Use of this source code is governed by the Apache 2.0 license that can be found in the  
license/LICENSE.txt file.\n \* \npackage kotlin.internal\n\n// a mod b (in arithmetical sense)\nprivate fun mod(a:  
Int, b: Int): Int {\n val mod = a % b\n return if (mod >= 0) mod else mod + b}\n\nprivate fun mod(a: Long, b:  
Long): Long {\n val mod = a % b\n return if (mod >= 0) mod else mod + b}\n\n// (a - b) mod c\nprivate fun  
differenceModulo(a: Int, b: Int, c: Int): Int {\n return mod(mod(a, c) - mod(b, c), c)\n}\n\nprivate fun  
differenceModulo(a: Long, b: Long, c: Long): Long {\n return mod(mod(a, c) - mod(b, c), c)\n}\n\n/\*\n \*  
Calculates the final element of a bounded arithmetic progression, i.e. the last element of the progression which is in  
the range\n \* from [start] to [end] in case of a positive [step], or from [end] to [start] in case of a negative\n \*  
[step].\n \* No validation on passed parameters is performed. The given parameters should satisfy the  
condition:\n \* - either `step > 0` and `start <= end`,\n \* - or `step < 0` and `start >= end`.\n \* @param start  
first element of the progression\n \* @param end ending bound for the progression\n \* @param step increment, or  
difference of successive elements in the progression\n \* @return the final element of the progression\n \*  
@suppress\n \* \n@PublishedApi\ninternal fun getProgressionLastElement(start: Int, end: Int, step: Int): Int = when  
{\n step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step)\n step < 0 -> if (start <= end)

```

end else end + differenceModulo(start, end, -step)\n else -> throw kotlin.IllegalArgumentException("Step is
zero.")\n}\n\n/**\n * Calculates the final element of a bounded arithmetic progression, i.e. the last element of the
progression which is in the range\n * from [start] to [end] in case of a positive [step], or from [end] to [start] in case
of a negative\n * [step].\n * No validation on passed parameters is performed. The given parameters should
satisfy the condition:\n * - either `step > 0` and `start <= end`,\n * - or `step < 0` and `start >= end`.\n *
@param start first element of the progression\n * @param end ending bound for the progression\n * @param step
increment, or difference of successive elements in the progression\n * @return the final element of the progression\n
* @suppress\n */\n@PublishedApi\ninternal fun getProgressionLastElement(start: Long, end: Long, step: Long):
Long = when {\n step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step)\n step < 0 -> if
(start <= end) end else end + differenceModulo(start, end, -step)\n else -> throw
kotlin.IllegalArgumentException("Step is zero.")\n}\n\n","/**\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\npackage kotlin.properties\nimport kotlin.reflect.KProperty\n\n/**\n * Standard property delegates.\n */\npublic object Delegates {\n /**\n * Returns a property delegate for a
read/write property with a non-`null` value that is initialized not during\n * object construction time but at a later
time. Trying to read the property before the initial value has been\n * assigned results in an exception.\n *
@sample samples.properties.Delegates.notNullDelegate\n */\n public fun <T : Any> notNull():
ReadWriteProperty<Any?, T> = NotNullVar()\n /**\n * Returns a property delegate for a read/write property
that calls a specified callback function when changed.\n * @param initialValue the initial value of the property.\n
* @param onChange the callback which is called after the change of the property is made. The value of the
property\n * has already been changed when this callback is invoked.\n *
@sample
samples.properties.Delegates.observableDelegate\n */\n public inline fun <T> observable(initialValue: T,
crossinline onChange: (property: KProperty<*>, oldValue: T, newValue: T) -> Unit):\n
ReadWriteProperty<Any?, T> =\n object : ObservableProperty<T>(initialValue) {\n override fun
afterChange(property: KProperty<*>, oldValue: T, newValue: T) = onChange(property, oldValue, newValue)\n
}\n\n /**\n * Returns a property delegate for a read/write property that calls a specified callback function when
changed,\n * allowing the callback to veto the modification.\n * @param initialValue the initial value of the
property.\n * @param onChange the callback which is called before a change to the property value is attempted.\n
* The value of the property hasn't been changed yet, when this callback is invoked.\n * If the callback returns
`true` the value of the property is being set to the new value,\n * and if the callback returns `false` the new value
is discarded and the property remains its old value.\n *
@sample
samples.properties.Delegates.vetoableDelegate\n * @sample
samples.properties.Delegates.throwVetoableDelegate\n */\n public inline fun <T> vetoable(initialValue: T,
crossinline onChange: (property: KProperty<*>, oldValue: T, newValue: T) -> Boolean):\n
ReadWriteProperty<Any?, T> =\n object : ObservableProperty<T>(initialValue) {\n override fun
beforeChange(property: KProperty<*>, oldValue: T, newValue: T): Boolean = onChange(property, oldValue,
newValue)\n }\n\n}\n\nprivate class NotNullVar<T : Any>(): ReadWriteProperty<Any?, T> {\n private var
value: T? = null\n public override fun getValue(thisRef: Any?, property: KProperty<*>): T {\n return value
?: throw IllegalStateException("Property ${property.name} should be initialized before get.")\n }\n\n public
override fun setValue(thisRef: Any?, property: KProperty<*>, value: T) {\n this.value = value\n
}\n}\n\n","/**\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\npackage kotlin.properties\nimport kotlin.reflect.KProperty\n\n/**\n * Base interface that can be used for
implementing property delegates of read-only properties.\n * This is provided only for convenience; you don't
have to extend this interface\n * as long as your property delegate has methods with the same signatures.\n *
@param T the type of object which owns the delegated property.\n * @param V the type of the property value.\n
*/\npublic fun interface ReadOnlyProperty<in T, out V> {\n /**\n * Returns the value of the property for the
given object.\n * @param thisRef the object for which the value is requested.\n * @param property the

```

```

metadata for the property.\n * @return the property value.\n */\n public operator fun getValue(thisRef: T,
property: KProperty<*>): V\n}\n\n/**\n * Base interface that can be used for implementing property delegates of
read-write properties.\n *\n * This is provided only for convenience; you don't have to extend this interface\n * as
long as your property delegate has methods with the same signatures.\n *\n * @param T the type of object which
owns the delegated property.\n * @param V the type of the property value.\n */\npublic interface
ReadWriteProperty<in T, V> : ReadOnlyProperty<T, V> {\n /**\n * Returns the value of the property for the
given object.\n * @param thisRef the object for which the value is requested.\n * @param property the
metadata for the property.\n * @return the property value.\n */\n public override operator fun
getValue(thisRef: T, property: KProperty<*>): V\n\n /**\n * Sets the value of the property for the given
object.\n * @param thisRef the object for which the value is requested.\n * @param property the metadata for
the property.\n * @param value the value to set.\n */\n public operator fun setValue(thisRef: T, property:
KProperty<*>, value: V)\n}\n\n/**\n * Base interface that can be used for implementing property delegate
providers.\n *\n * This is provided only for convenience; you don't have to extend this interface\n * as long as your
delegate provider has a method with the same signature.\n *\n * @param T the type of object which owns the
delegated property.\n * @param D the type of property delegates this provider provides.\n
*/\n\n@SinceKotlin("1.4")\npublic fun interface PropertyDelegateProvider<in T, out D> {\n /**\n * Returns the
delegate of the property for the given object.\n *\n * This function can be used to extend the logic of creating
the object (e.g. perform validation checks)\n * to which the property implementation is delegated.\n *\n *
@param thisRef the object for which property delegate is requested.\n * @param property the metadata for the
property.\n * @return the property delegate.\n */\n public operator fun provideDelegate(thisRef: T, property:
KProperty<*>): D\n}\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.properties\n\nimport kotlin.reflect.KProperty\n\n/**\n *
Implements the core logic of a property delegate for a read/write property that calls callback functions when
changed.\n * @param initialValue the initial value of the property.\n */\npublic abstract class
ObservableProperty<V>(initialValue: V) : ReadWriteProperty<Any?, V> {\n private var value = initialValue\n\n
/**\n * The callback which is called before a change to the property value is attempted.\n * The value of the
property hasn't been changed yet, when this callback is invoked.\n * If the callback returns `true` the value of the
property is being set to the new value,\n * and if the callback returns `false` the new value is discarded and the
property remains its old value.\n */\n protected open fun beforeChange(property: KProperty<*>, oldValue: V,
newValue: V): Boolean = true\n\n /**\n * The callback which is called after the change of the property is made.
The value of the property\n * has already been changed when this callback is invoked.\n */\n protected open
fun afterChange(property: KProperty<*>, oldValue: V, newValue: V): Unit {}\n\n public override fun
getValue(thisRef: Any?, property: KProperty<*>): V {\n return value\n }\n\n public override fun
setValue(thisRef: Any?, property: KProperty<*>, value: V) {\n val oldValue = this.value\n if
(!beforeChange(property, oldValue, value)) {\n return\n }\n this.value = value\n
afterChange(property, oldValue, value)\n }\n}\n\n"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\n@file:Suppress("PackageDirectoryMismatch")\npackage
kotlin\n\nimport kotlin.reflect.*\n\n/**\n * An extension operator that allows delegating a read-only property of type
[V]\n * to a property reference to a property of type [V] or its subtype.\n *\n * @receiver A property reference to a
read-only or mutable property of type [V] or its subtype.\n * The reference is without a receiver, i.e. it either
references a top-level property or\n * has the receiver bound to it.\n *\n * Example:\n *\n * ```\n * class Login(val
username: String)\n * val defaultLogin = Login("Admin")\n * val defaultUsername by defaultLogin::username\n *
// equivalent to\n * val defaultUserName get() = defaultLogin.username\n * ```\n
*/\n\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline operator fun <V>
KProperty0<V>.getValue(thisRef: Any?, property: KProperty<*>): V {\n return get()\n}\n\n/**\n * An extension
operator that allows delegating a mutable property of type [V]\n * to a property reference to a mutable property of

```

the same type [V].\n \* \n \* @receiver A property reference to a mutable property of type [V].\n \* The reference is without a receiver, i.e. it either references a top-level property or\n \* has the receiver bound to it.\n \* \n \* Example:\n \* \n \* ```\n \* class Login(val username: String, var incorrectAttemptCounter: Int = 0)\n \* val defaultLogin = Login("Admin")\n \* var defaultLoginAttempts by defaultLogin::incorrectAttemptCounter\n \* // equivalent to\n \* var defaultLoginAttempts: Int\n \* get() = defaultLogin.incorrectAttemptCounter\n \* set(value) { defaultLogin.incorrectAttemptCounter = value }\n \* ```\n \* \n \* \n \* @SinceKotlin("1.4")\n \* @kotlin.internal.InlineOnly\n \* public inline operator fun <V> KMutableProperty0<V>.setValue(thisRef: Any?, property: KProperty<\*>, value: V) {\n \* set(value)\n \* }\n \* \n \* An extension operator that allows delegating a read-only member or extension property of type [V]\n \* to a property reference to a member or extension property of type [V] or its subtype.\n \* \n \* @receiver A property reference to a read-only or mutable property of type [V] or its subtype.\n \* The reference has an unbound receiver of type [T].\n \* \n \* Example:\n \* \n \* ```\n \* class Login(val username: String)\n \* val Login.user by Login::username\n \* // equivalent to\n \* val Login.user get() = this.username\n \* ```\n \* \n \* \n \* @SinceKotlin("1.4")\n \* @kotlin.internal.InlineOnly\n \* public inline operator fun <T, V> KProperty1<T, V>.getValue(thisRef: T, property: KProperty<\*>): V {\n \* return get(thisRef)\n \* }\n \* \n \* An extension operator that allows delegating a mutable member or extension property of type [V]\n \* to a property reference to a member or extension mutable property of the same type [V].\n \* \n \* @receiver A property reference to a read-only or mutable property of type [V] or its subtype.\n \* The reference has an unbound receiver of type [T].\n \* \n \* Example:\n \* \n \* ```\n \* class Login(val username: String, var incorrectAttemptCounter: Int)\n \* var Login.attempts by Login::incorrectAttemptCounter\n \* // equivalent to\n \* var Login.attempts: Int\n \* get() = this.incorrectAttemptCounter\n \* set(value) { this.incorrectAttemptCounter = value }\n \* ```\n \* \n \* \n \* @SinceKotlin("1.4")\n \* @kotlin.internal.InlineOnly\n \* public inline operator fun <T, V> KMutableProperty1<T, V>.setValue(thisRef: T, property: KProperty<\*>, value: V) {\n \* set(thisRef, value)\n \* }, /\*\n \* Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n \* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n \* \n \* \n \* package kotlin.random\n \* \n \* import kotlin.math.nextDown\n \* \n \* /\*\*\n \* An abstract class that is implemented by random number generator algorithms.\n \* \n \* The companion object [Random.Default] is the default instance of [Random].\n \* \n \* To get a seeded instance of random generator use [Random] function.\n \* \n \* @sample samples.random.Randoms.defaultRandom\n \* \n \* @SinceKotlin("1.3")\n \* public abstract class Random {\n \* \n \* /\*\*\n \* Gets the next random [bitCount] number of bits.\n \* \n \* Generates an `Int` whose lower [bitCount] bits are filled with random values and the remaining upper bits are zero.\n \* \n \* @param bitCount number of bits to generate, must be in range 0..32, otherwise the behavior is unspecified.\n \* \n \* @sample samples.random.Randoms.nextBits\n \* \n \* ^\n \* public abstract fun nextBits(bitCount: Int): Int\n \* /\*\*\n \* Gets the next random `Int` from the random number generator.\n \* \n \* Generates an `Int` random value uniformly distributed between `Int.MIN\_VALUE` and `Int.MAX\_VALUE` (inclusive).\n \* \n \* @sample samples.random.Randoms.nextInt\n \* \n \* ^\n \* public open fun nextInt(): Int = nextBits(32)\n \* /\*\*\n \* Gets the next random non-negative `Int` from the random number generator less than the specified [until] bound.\n \* \n \* Generates an `Int` random value uniformly distributed between `0` (inclusive) and the specified [until] bound (exclusive).\n \* \n \* @param until must be positive.\n \* \n \* @throws IllegalArgumentException if [until] is negative or zero.\n \* \n \* @sample samples.random.Randoms.nextIntFromUntil\n \* \n \* ^\n \* public open fun nextInt(until: Int): Int = nextInt(0, until)\n \* /\*\*\n \* Gets the next random `Int` from the random number generator in the specified range.\n \* \n \* Generates an `Int` random value uniformly distributed between the specified [from] (inclusive) and [until] (exclusive) bounds.\n \* \n \* @throws IllegalArgumentException if [from] is greater than or equal to [until].\n \* \n \* @sample samples.random.Randoms.nextIntFromUntil\n \* \n \* ^\n \* public open fun nextInt(from: Int, until: Int): Int {\n \* checkRangeBounds(from, until)\n \* val n = until - from\n \* if (n > 0 || n == Int.MIN\_VALUE) {\n \* val rnd = if (n and -n == n) {\n \* val bitCount = fastLog2(n)\n \* nextBits(bitCount)\n \* } else {\n \* var v: Int\n \* do {\n \* val bits = nextInt().ushr(1)\n \* v = bits % n\n \* } while (bits - v + (n - 1) < 0)\n \* v\n \* } return from + rnd\n \* } else {\n \* while



```

(true) {\n val rnd = nextInt()\n if (rnd in from until until) return rnd\n }\n }\n }\n\n/**\n * Gets the next random `Long` from the random number generator.\n *\n * Generates a `Long` random value uniformly distributed between `Long.MIN_VALUE` and `Long.MAX_VALUE` (inclusive).\n *\n * @sample samples.random.Randoms.nextLong\n *\n public open fun nextLong(): Long = nextInt().toLong().shl(32) + nextInt()\n\n /**\n * Gets the next random non-negative `Long` from the random number generator less than the specified [until] bound.\n *\n * Generates a `Long` random value uniformly distributed between `0` (inclusive) and the specified [until] bound (exclusive).\n *\n * @param until must be positive.\n *\n * @throws IllegalArgumentException if [until] is negative or zero.\n *\n * @sample samples.random.Randoms.nextLongFromUntil\n *\n public open fun nextLong(until: Long): Long = nextLong(0, until)\n\n /**\n * Gets the next random `Long` from the random number generator in the specified range.\n *\n * Generates a `Long` random value uniformly distributed between the specified [from] (inclusive) and [until] (exclusive) bounds.\n *\n * @throws IllegalArgumentException if [from] is greater than or equal to [until].\n *\n * @sample samples.random.Randoms.nextLongFromUntil\n *\n public open fun nextLong(from: Long, until: Long): Long {\n checkRangeBounds(from, until)\n val n = until - from\n if (n > 0) {\n val rnd: Long\n if (n and -n == n) {\n val nLow = n.toInt()\n val nHigh = (n ushr 32).toInt()\n rnd = when {\n nLow != 0 -> {\n val bitCount = fastLog2(nLow)\n // toUInt().toLong()\n nextBits(bitCount).toLong() and 0xFFFF_FFFF\n }\n nHigh == 1 -> {\n // toUInt().toLong()\n nextInt().toLong() and 0xFFFF_FFFF\n }\n else -> {\n val bitCount = fastLog2(nHigh)\n nextBits(bitCount).toLong().shl(32) + (nextInt().toLong() and 0xFFFF_FFFF)\n }\n }\n } else {\n var v: Long\n do {\n val bits = nextLong().ushr(1)\n v = bits % n\n } while (bits - v + (n - 1) < 0)\n rnd = v\n }\n return from + rnd\n } else {\n while (true) {\n val rnd = nextLong()\n if (rnd in from until until) return rnd\n }\n }\n }\n\n /**\n * Gets the next random [Boolean] value.\n *\n * @sample samples.random.Randoms.nextBoolean\n *\n public open fun nextBoolean(): Boolean = nextBits(1) != 0\n\n /**\n * Gets the next random [Double] value uniformly distributed between 0 (inclusive) and 1 (exclusive).\n *\n * @sample samples.random.Randoms.nextDouble\n *\n public open fun nextDouble(): Double = doubleFromParts(nextBits(26), nextBits(27))\n\n /**\n * Gets the next random non-negative `Double` from the random number generator less than the specified [until] bound.\n *\n * Generates a `Double` random value uniformly distributed between 0 (inclusive) and [until] (exclusive).\n *\n * @throws IllegalArgumentException if [until] is negative or zero.\n *\n * @sample samples.random.Randoms.nextDoubleFromUntil\n *\n public open fun nextDouble(until: Double): Double = nextDouble(0.0, until)\n\n /**\n * Gets the next random `Double` from the random number generator in the specified range.\n *\n * Generates a `Double` random value uniformly distributed between the specified [from] (inclusive) and [until] (exclusive) bounds.\n *\n * [from] and [until] must be finite otherwise the behavior is unspecified.\n *\n * @throws IllegalArgumentException if [from] is greater than or equal to [until].\n *\n * @sample samples.random.Randoms.nextDoubleFromUntil\n *\n public open fun nextDouble(from: Double, until: Double): Double {\n checkRangeBounds(from, until)\n val size = until - from\n val r = if (size.isInfinite() && from.isFinite() && until.isFinite()) {\n val r1 = nextDouble() * (until / 2 - from / 2)\n from + r1 + r1\n } else {\n from + nextDouble() * size\n }\n return if (r >= until) until.nextDown() else r\n }\n\n /**\n * Gets the next random [Float] value uniformly distributed between 0 (inclusive) and 1 (exclusive).\n *\n * @sample samples.random.Randoms.nextFloat\n *\n public open fun nextFloat(): Float = nextBits(24) / (1 shl 24).toFloat()\n\n /**\n * Fills a subrange of the specified byte [array] starting from [fromIndex] inclusive and ending [toIndex] exclusive\n * with random bytes.\n *\n * @return [array] with the subrange filled with random bytes.\n *\n * @sample samples.random.Randoms.nextBytes\n *\n public open fun nextBytes(array: ByteArray, fromIndex: Int = 0, toIndex: Int = array.size): ByteArray {\n require(fromIndex in 0..array.size && toIndex in 0..array.size) { \"fromIndex ($fromIndex) or toIndex ($toIndex) are out of range: 0..${array.size}.\" }\n require(fromIndex <= toIndex) { \"fromIndex ($fromIndex) must be not greater than

```

```

toIndex ($toIndex).` } \n \n val steps = (toIndex - fromIndex) / 4 \n \n var position = fromIndex \n
repeat(steps) { \n val v = nextInt() \n array[position] = v.toByte() \n array[position + 1] =
v.ushr(8).toByte() \n array[position + 2] = v.ushr(16).toByte() \n array[position + 3] =
v.ushr(24).toByte() \n position += 4 \n } \n \n val remainder = toIndex - position \n val vr =
nextInt(remainder * 8) \n for (i in 0 until remainder) { \n array[position + i] = vr.ushr(i * 8).toByte() \n
 } \n \n return array \n } \n \n /** \n * Fills the specified byte [array] with random bytes and returns it. \n * \n
* @return [array] filled with random bytes. \n * \n * @sample samples.random.Randoms.nextBytes \n */ \n
public open fun nextBytes(array: ByteArray): ByteArray = nextBytes(array, 0, array.size) \n \n /** \n * Creates a
byte array of the specified [size], filled with random bytes. \n * \n * @sample
samples.random.Randoms.nextBytes \n */ \n public open fun nextBytes(size: Int): ByteArray =
nextBytes(ByteArray(size)) \n \n /** \n * The default random number generator. \n * \n * On JVM this
generator is thread-safe, its methods can be invoked from multiple threads. \n * \n * @sample
samples.random.Randoms.defaultRandom \n */ \n companion object Default : Random(), Serializable { \n
private val defaultRandom: Random = defaultPlatformRandom() \n private object Serialized : Serializable { \n
 private const val serialVersionUID = 0L \n \n private fun readResolve(): Any = Random \n } \n \n
private fun writeReplace(): Any = Serialized \n \n override fun nextBits(bitCount: Int): Int =
defaultRandom.nextBits(bitCount) \n override fun nextInt(): Int = defaultRandom.nextInt() \n override fun
nextInt(until: Int): Int = defaultRandom.nextInt(until) \n override fun nextInt(from: Int, until: Int): Int =
defaultRandom.nextInt(from, until) \n \n override fun nextLong(): Long = defaultRandom.nextLong() \n
override fun nextLong(until: Long): Long = defaultRandom.nextLong(until) \n override fun nextLong(from:
Long, until: Long): Long = defaultRandom.nextLong(from, until) \n \n override fun nextBoolean(): Boolean =
defaultRandom.nextBoolean() \n \n override fun nextDouble(): Double = defaultRandom.nextDouble() \n
override fun nextDouble(until: Double): Double = defaultRandom.nextDouble(until) \n override fun
nextDouble(from: Double, until: Double): Double = defaultRandom.nextDouble(from, until) \n \n override fun
nextFloat(): Float = defaultRandom.nextFloat() \n \n override fun nextBytes(array: ByteArray): ByteArray =
defaultRandom.nextBytes(array) \n override fun nextBytes(size: Int): ByteArray =
defaultRandom.nextBytes(size) \n override fun nextBytes(array: ByteArray, fromIndex: Int, toIndex: Int):
ByteArray = \n defaultRandom.nextBytes(array, fromIndex, toIndex) \n } \n } \n \n /** \n * Returns a repeatable
random number generator seeded with the given [seed] `Int` value. \n * \n * Two generators with the same seed
produce the same sequence of values within the same version of Kotlin runtime. \n * \n * *Note:* Future versions of
Kotlin may change the algorithm of this seeded number generator so that it will return \n * a sequence of values
different from the current one for a given seed. \n * \n * On JVM the returned generator is NOT thread-safe. Do not
invoke it from multiple threads without proper synchronization. \n * \n * @sample
samples.random.Randoms.seededRandom \n */ \n @SinceKotlin("1.3") \n public fun Random(seed: Int): Random =
XorWowRandom(seed, seed.shr(31)) \n \n /** \n * Returns a repeatable random number generator seeded with the
given [seed] `Long` value. \n * \n * Two generators with the same seed produce the same sequence of values within
the same version of Kotlin runtime. \n * \n * *Note:* Future versions of Kotlin may change the algorithm of this
seeded number generator so that it will return \n * a sequence of values different from the current one for a given
seed. \n * \n * On JVM the returned generator is NOT thread-safe. Do not invoke it from multiple threads without
proper synchronization. \n * \n * @sample samples.random.Randoms.seededRandom \n
*/ \n @SinceKotlin("1.3") \n public fun Random(seed: Long): Random = XorWowRandom(seed.toInt(),
seed.shr(32).toInt()) \n \n /** \n * Gets the next random `Int` from the random number generator in the specified
[range]. \n * \n * Generates an `Int` random value uniformly distributed in the specified [range]: \n * from `range.start`
inclusive to `range.endInclusive` inclusive. \n * \n * @throws IllegalArgumentException if [range] is empty. \n
*/ \n @SinceKotlin("1.3") \n public fun Random.nextInt(range: IntRange): Int = when { \n range.isEmpty() -> throw
IllegalArgumentException("Cannot get random in empty range: $range") \n range.last < Int.MAX_VALUE ->
nextInt(range.first, range.last + 1) \n range.first > Int.MIN_VALUE -> nextInt(range.first - 1, range.last) + 1 \n
else -> nextInt() \n } \n \n /** \n * Gets the next random `Long` from the random number generator in the specified

```

```

[range].\n * Generates a `Long` random value uniformly distributed in the specified [range]:\n * from
`range.start` inclusive to `range.endInclusive` inclusive.\n * @throws IllegalArgumentException if [range] is
empty.\n * @SinceKotlin("1.3")\npublic fun Random.nextLong(range: LongRange): Long = when {\n
range.isEmpty() -> throw IllegalArgumentException("Cannot get random in empty range: $range")\n
range.last < Long.MAX_VALUE -> nextLong(range.first, range.last + 1)\n
range.first > Long.MIN_VALUE ->
nextLong(range.first - 1, range.last) + 1\n
else -> nextLong()\n}\n\ninternal expect fun
defaultPlatformRandom(): Random\n\ninternal expect fun doubleFromParts(hi26: Int, low27: Int): Double\n\ninternal
fun fastLog2(value: Int): Int = 31 - value.countLeadingZeroBits()\n\n/** Takes upper [bitCount] bits (0..32) from
this number. */\n\ninternal fun Int.takeUpperBits(bitCount: Int): Int =\n
this.ushr(32 - bitCount) and (-
bitCount).shr(31)\n\ninternal fun checkRangeBounds(from: Int, until: Int) = require(until > from) {
boundsErrorMessage(from, until) }\n\ninternal fun checkRangeBounds(from: Long, until: Long) = require(until >
from) { boundsErrorMessage(from, until) }\n\ninternal fun checkRangeBounds(from: Double, until: Double) =
require(until > from) { boundsErrorMessage(from, until) }\n\ninternal fun boundsErrorMessage(from: Any, until:
Any) = "Random range is empty: [$from, $until]."\n\n"/**\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\npackage kotlin.random\n\n/**\n * Gets the next random [UInt]
from the random number generator.\n * Generates a [UInt] random value uniformly distributed between
[UInt.MIN_VALUE] and [UInt.MAX_VALUE] (inclusive).\n
*\n * @SinceKotlin("1.5")\n * @WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextUInt(): UInt = nextInt().toUInt()\n\n/**\n * Gets the next random [UInt] from the random number
generator less than the specified [until] bound.\n * Generates a [UInt] random value uniformly distributed
between `0` (inclusive) and the specified [until] bound (exclusive).\n * @throws IllegalArgumentException if
[until] is zero.\n * @SinceKotlin("1.5")\n * @WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextUInt(until: UInt): UInt = nextUInt(0u, until)\n\n/**\n * Gets the next random [UInt] from the random
number generator in the specified range.\n * Generates a [UInt] random value uniformly distributed between the
specified [from] (inclusive) and [until] (exclusive) bounds.\n * @throws IllegalArgumentException if [from] is
greater than or equal to [until].\n
*\n * @SinceKotlin("1.5")\n * @WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextUInt(from: UInt, until: UInt): UInt {\n
checkUIntRangeBounds(from, until)\n\n
val signedFrom = from.toInt() xor Int.MIN_VALUE\n
val signedUntil = until.toInt() xor Int.MIN_VALUE\n\n
val signedResult = nextInt(signedFrom, signedUntil) xor Int.MIN_VALUE\n
return signedResult.toUInt()\n}\n\n/**\n * Gets the next
random [UInt] from the random number generator in the specified [range].\n * Generates a [UInt] random value
uniformly distributed in the specified [range]:\n * from `range.start` inclusive to `range.endInclusive` inclusive.\n
*\n * @throws IllegalArgumentException if [range] is empty.\n
*\n * @SinceKotlin("1.5")\n * @WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextUInt(range: UIntRange): UInt = when {\n
range.isEmpty() -> throw
IllegalArgumentException("Cannot get random in empty range: $range")\n
range.last < UInt.MAX_VALUE ->
nextUInt(range.first, range.last + 1u)\n
range.first > UInt.MIN_VALUE -> nextUInt(range.first - 1u, range.last) +
1u\n
else -> nextUInt()\n}\n\n/**\n * Gets the next random [ULong] from the random number generator.\n *
Generates a [ULong] random value uniformly distributed between [ULong.MIN_VALUE] and
[ULong.MAX_VALUE] (inclusive).\n
*\n * @SinceKotlin("1.5")\n * @WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextULong(): ULong = nextLong().toULong()\n\n/**\n * Gets the next random [ULong] from the random
number generator less than the specified [until] bound.\n * Generates a [ULong] random value uniformly
distributed between `0` (inclusive) and the specified [until] bound (exclusive).\n * @throws
IllegalArgumentException if [until] is zero.\n
*\n * @SinceKotlin("1.5")\n * @WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextULong(until: ULong): ULong = nextULong(0uL, until)\n\n/**\n * Gets the next random [ULong] from

```

```

the random number generator in the specified range.\n * \n * Generates a [ULong] random value uniformly
distributed between the specified [from] (inclusive) and [until] (exclusive) bounds.\n * \n * @throws
IllegalArgumentExcepion if [from] is greater than or equal to [until].\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextULong(from: ULong, until: ULong): ULong {\n checkULongRangeBounds(from, until)\n\n val
signedFrom = from.toLong() xor Long.MIN_VALUE\n val signedUntil = until.toLong() xor
Long.MIN_VALUE\n\n val signedResult = nextLong(signedFrom, signedUntil) xor Long.MIN_VALUE\n
return signedResult.toULong()\n}\n\n/**\n * Gets the next random [ULong] from the random number generator in
the specified [range].\n * \n * Generates a [ULong] random value uniformly distributed in the specified [range]:\n *
from `range.start` inclusive to `range.endInclusive` inclusive.\n * \n * @throws IllegalArgumentExcepion if [range]
is empty.\n * \n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextULong(range: ULongRange): ULong = when {\n range.isEmpty() -> throw
IllegalArgumentExcepion("Cannot get random in empty range: $range")\n range.last < ULong.MAX_VALUE -
> nextULong(range.first, range.last + 1u)\n range.first > ULong.MIN_VALUE -> nextULong(range.first - 1u,
range.last) + 1u\n else -> nextULong()\n}\n\n/**\n * Fills the specified unsigned byte [array] with random bytes
and returns it.\n * \n * @return [array] filled with random bytes.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Random.nextUBytes(array: UByteArray):
UByteArray {\n nextBytes(array.asByteArray())\n return array\n}\n\n/**\n * Creates an unsigned byte array of
the specified [size], filled with random bytes.\n * \n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun Random.nextUBytes(size: Int): UByteArray = nextBytes(size).asUByteArray()\n}\n\n/**\n * Fills a subrange of the
specified `UByte` [array] starting from [fromIndex] inclusive and ending [toIndex] exclusive with random
UBytes.\n * \n * @return [array] with the subrange filled with random bytes.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Random.nextUBytes(array: UByteArray,
fromIndex: Int = 0, toIndex: Int = array.size): UByteArray {\n nextBytes(array.asByteArray(), fromIndex,
toIndex)\n return array\n}\n\n\ninternal fun checkUIntRangeBounds(from: UInt, until: UInt) = require(until >
from) { boundsErrorMessage(from, until) }\ninternal fun checkULongRangeBounds(from: ULong, until: ULong) =
require(until > from) { boundsErrorMessage(from, until) }\n", /*\n * Copyright 2010-2018 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n * \n\npackage kotlin.random\n\n/**\n * Random number generator,
using Marsaglia's "xorwow" algorithm\n * \n * Cycles after 2^192 - 2^32 repetitions.\n * \n * For more details, see
Marsaglia, George (July 2003). "Xorshift RNGs". Journal of Statistical Software. 8 (14).
doi:10.18637/jss.v008.i14\n * \n * Available at https://www.jstatsoft.org/v08/i14/paper\n * \n *\ninternal class
XorWowRandom internal constructor(\n private var x: Int,\n private var y: Int,\n private var z: Int,\n private
var w: Int,\n private var v: Int,\n private var addend: Int\n) : Random(), Serializable {\n\n internal
constructor(seed1: Int, seed2: Int) :n this(seed1, seed2, 0, 0, seed1.inv(), (seed1 shl 10) xor (seed2 ushr
4))\n\n init {\n require((x or y or z or w or v) != 0) { "Initial state must have at least one non-zero element."
}\n\n // some trivial seeds can produce several values with zeroes in upper bits, so we discard first 64\n
repeat(64) { nextInt() }\n }\n\n override fun nextInt(): Int {\n // Equivalent to the xorwow algorithm\n //
From Marsaglia, G. 2003. Xorshift RNGs. J. Statis. Soft. 8, 14, p. 5\n var t = x\n t = t xor (t ushr 2)\n x
= y\n y = z\n z = w\n val v0 = v\n w = v0\n t = (t xor (t shl 1)) xor v0 xor (v0 shl 4)\n v =
t\n addend += 362437\n return t + addend\n }\n\n override fun nextBits(bitCount: Int): Int =\n
nextInt().takeUpperBits(bitCount)\n}\n\n private companion object {\n private const val serialVersionUID: Long
= 0L\n }\n}\n", /*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n *
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin.ranges\n\n/**\n * An iterator over a progression of
values of type `Char`.\n * \n * @property step the number by which the value is incremented on each step.\n * \n\ninternal
class CharProgressionIterator(first: Char, last: Char, val step: Int) : CharIterator() {\n private val finalElement: Int
= last.code\n private var hasNext: Boolean = if (step > 0) first <= last else first >= last\n private var next: Int = if

```

```

(hasNext) first.code else finalElement\n\n override fun hasNext(): Boolean = hasNext\n\n override fun
nextChar(): Char {\n val value = next\n if (value == finalElement) {\n if (!hasNext) throw
kotlin.NoSuchElementException()\n hasNext = false\n }\n else {\n next += step\n }\n
return value.toChar()\n }\n}\n\n/**\n * An iterator over a progression of values of type `Int`. \n * @property step
the number by which the value is incremented on each step.\n */\n\ninternal class IntProgressionIterator(first: Int, last:
Int, val step: Int) : IntIterator() {\n private val finalElement: Int = last\n private var hasNext: Boolean = if (step >
0) first <= last else first >= last\n private var next: Int = if (hasNext) first else finalElement\n\n override fun
hasNext(): Boolean = hasNext\n\n override fun nextInt(): Int {\n val value = next\n if (value ==
finalElement) {\n if (!hasNext) throw kotlin.NoSuchElementException()\n hasNext = false\n }\n
else {\n next += step\n }\n return value\n }\n}\n\n/**\n * An iterator over a progression of values
of type `Long`. \n * @property step the number by which the value is incremented on each step.\n */\n\ninternal class
LongProgressionIterator(first: Long, last: Long, val step: Long) : LongIterator() {\n private val finalElement: Long
= last\n private var hasNext: Boolean = if (step > 0) first <= last else first >= last\n private var next: Long = if
(hasNext) first else finalElement\n\n override fun hasNext(): Boolean = hasNext\n\n override fun nextLong():
Long {\n val value = next\n if (value == finalElement) {\n if (!hasNext) throw
kotlin.NoSuchElementException()\n hasNext = false\n }\n else {\n next += step\n }\n
return value\n }\n}\n\n"/**\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin.ranges\n\nimport
kotlin.internal.getProgressionLastElement\n\n/**\n * A progression of values of type `Char`. \n */\n\npublic open class
CharProgression\n internal constructor\n (\n start: Char,\n endInclusive: Char,\n step: Int\n
) : Iterable<Char> {\n init {\n if (step == 0) throw kotlin.IllegalArgumentException("\u201cStep must be non-
zero.\u201c")\n if (step == Int.MIN_VALUE) throw kotlin.IllegalArgumentException("\u201cStep must be greater than
Int.MIN_VALUE to avoid overflow on negation.\u201c")\n }\n\n /**\n * The first element in the progression.\n
*\n */\n public val first: Char = start\n\n /**\n * The last element in the progression.\n
*\n */\n public val last: Char = getProgressionLastElement(start.code, endInclusive.code, step).toChar()\n\n /**\n * The step of the
progression.\n
*\n */\n public val step: Int = step\n\n override fun iterator(): CharIterator =
CharProgressionIterator(first, last, step)\n\n /**\n * Checks if the progression is empty.\n
*\n */\n * Progression with a positive step is empty if its first element is greater than the last element.\n
*\n * Progression with a negative step is empty if its first element is less than the last element.\n
*\n */\n public open fun isEmpty(): Boolean = if
(step > 0) first > last else first < last\n\n override fun equals(other: Any?): Boolean =\n other is
CharProgression && (isEmpty() && other.isEmpty()) ||\n first == other.first && last == other.last && step ==
other.step)\n\n override fun hashCode(): Int =\n if (isEmpty()) -1 else (31 * (31 * first.code + last.code) +
step)\n\n override fun toString(): String = if (step > 0) "\u201c$first..$last step $step\u201c" else "\u201c$first downTo $last step ${-
step}\u201c"\n\n companion object {\n /**\n * Creates CharProgression within the specified bounds of a
closed range.\n
*\n */\n * The progression starts with the [rangeStart] value and goes toward the [rangeEnd]
value not excluding it, with the specified [step].\n
*\n * In order to go backwards the [step] must be negative.\n
*\n * [step] must be greater than `Int.MIN_VALUE` and not equal to zero.\n
*\n */\n public fun
fromClosedRange(rangeStart: Char, rangeEnd: Char, step: Int): CharProgression = CharProgression(rangeStart,
rangeEnd, step)\n }\n}\n\n/**\n * A progression of values of type `Int`. \n */\n\npublic open class IntProgression\n
internal constructor\n (\n start: Int,\n endInclusive: Int,\n step: Int\n
) : Iterable<Int> {\n init {\n if (step == 0) throw kotlin.IllegalArgumentException("\u201cStep must be non-zero.\u201c")\n
if (step == Int.MIN_VALUE) throw kotlin.IllegalArgumentException("\u201cStep must be greater than Int.MIN_VALUE to avoid
overflow on negation.\u201c")\n }\n\n /**\n * The first element in the progression.\n
*\n */\n public val first: Int = start\n\n /**\n * The last element in the progression.\n
*\n */\n public val last: Int =
getProgressionLastElement(start, endInclusive, step)\n\n /**\n * The step of the progression.\n
*\n */\n public val step: Int = step\n\n override fun iterator(): IntIterator = IntProgressionIterator(first, last, step)\n\n /**\n *
Checks if the progression is empty.\n
*\n */\n * Progression with a positive step is empty if its first element is

```

```

greater than the last element.\n * Progression with a negative step is empty if its first element is less than the last
element.\n */\n public open fun isEmpty(): Boolean = if (step > 0) first > last else first < last\n\n override fun
equals(other: Any?): Boolean =\n other is IntProgression && (isEmpty() && other.isEmpty()) ||\n first ==
other.first && last == other.last && step == other.step)\n\n override fun hashCode(): Int =\n if (isEmpty()) -1
else (31 * (31 * first + last) + step)\n\n override fun toString(): String = if (step > 0) \"$first..$last step $step\" else
\"$first downTo $last step $-step\"\n\n companion object {\n /**\n * Creates IntProgression within the
specified bounds of a closed range.\n *\n * The progression starts with the [rangeStart] value and goes
toward the [rangeEnd] value not excluding it, with the specified [step].\n *\n * In order to go backwards the [step]
must be negative.\n *\n * [step] must be greater than `Int.MIN_VALUE` and not equal to zero.\n */\n public fun fromClosedRange(rangeStart: Int, rangeEnd: Int, step: Int): IntProgression =
IntProgression(rangeStart, rangeEnd, step)\n }\n\n /**\n * A progression of values of type `Long`.\n */\n public
open class LongProgression\n internal constructor(\n start: Long,\n endInclusive: Long,\n step: Long\n) : Iterable<Long> {\n init {\n if (step == 0L) throw kotlin.IllegalArgumentException(\"Step
must be non-zero.\")\n if (step == Long.MIN_VALUE) throw kotlin.IllegalArgumentException(\"Step must be
greater than Long.MIN_VALUE to avoid overflow on negation.\")\n }\n\n /**\n * The first element in the
progression.\n */\n public val first: Long = start\n\n /**\n * The last element in the progression.\n */\n public val last: Long = getProgressionLastElement(start, endInclusive, step)\n\n /**\n * The step of the
progression.\n */\n public val step: Long = step\n\n override fun iterator(): LongIterator =
LongProgressionIterator(first, last, step)\n\n /**\n * Checks if the progression is empty.\n */\n *
Progression with a positive step is empty if its first element is greater than the last element.\n * Progression with a
negative step is empty if its first element is less than the last element.\n */\n public open fun isEmpty(): Boolean
= if (step > 0) first > last else first < last\n\n override fun equals(other: Any?): Boolean =\n other is
LongProgression && (isEmpty() && other.isEmpty()) ||\n first == other.first && last == other.last && step ==
other.step)\n\n override fun hashCode(): Int =\n if (isEmpty()) -1 else (31 * (31 * (first xor (first ushr 32)) +
(last xor (last ushr 32))) + (step xor (step ushr 32))).toInt()\n\n override fun toString(): String = if (step > 0)
\"$first..$last step $step\" else \"$first downTo $last step $-step\"\n\n companion object {\n /**\n *
Creates LongProgression within the specified bounds of a closed range.\n *\n * The progression starts
with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].\n *
In order to go backwards the [step] must be negative.\n *\n * [step] must be greater than
`Long.MIN_VALUE` and not equal to zero.\n */\n public fun fromClosedRange(rangeStart: Long,
rangeEnd: Long, step: Long): LongProgression = LongProgression(rangeStart, rangeEnd, step)\n }\n\n /**\n * Represents a range of values (for example, numbers or characters) where both the lower and
upper bounds are included in the range.\n * See the [Kotlin language
documentation](https://kotlinlang.org/docs/reference/ranges.html) for more information.\n */\n public interface
ClosedRange<T : Comparable<T>> {\n /**\n * The minimum value in the range.\n */\n public val start:
T\n\n /**\n * The maximum value in the range (inclusive).\n */\n public val endInclusive: T\n\n /**\n * Checks whether the specified [value] belongs to the range.\n */\n * A value belongs to the closed range if it is
greater than or equal to the [start] bound and less than or equal to the [endInclusive] bound.\n */\n public
operator fun contains(value: T): Boolean = value >= start && value <= endInclusive\n\n /**\n * Checks
whether the range is empty.\n */\n * The range is empty if its start value is greater than the end value.\n */\n public fun isEmpty(): Boolean = start > endInclusive\n }\n\n /**\n * Represents a range of values (for example,
numbers or characters) where the upper bound is not included in the range.\n * See the [Kotlin language
documentation](https://kotlinlang.org/docs/reference/ranges.html) for more information.\n */\n *
\n * @SinceKotlin("1.7")\n * @ExperimentalStdlibApi\n public interface OpenEndRange<T : Comparable<T>> {\n
 /**\n * The minimum value in the range.\n */\n public val start: T\n\n /**\n * The maximum value in the
range (exclusive).\n */\n * @throws IllegalStateException can be thrown if the exclusive end bound cannot be

```

represented  
 \* with a value of type [T].  
 \*  
 public val endExclusive: T  
 /\*\*  
 \* Checks whether the specified [value] belongs to the range.  
 \*  
 \* A value belongs to the open-ended range if it is greater than or equal to the [start] bound and strictly less than the [endExclusive] bound.  
 \*  
 public operator fun contains(value: T): Boolean = value >= start && value < endExclusive  
 /\*\*  
 \* Checks whether the range is empty.  
 \*  
 \* The open-ended range is empty if its start value is greater than or equal to the end value.  
 \*  
 public fun isEmpty(): Boolean = start >= endExclusive  
 }  
 /\*\*  
 \* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.  
 \* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

```

*

@file:kotlin.jvm.JvmMultifileClass

@file:kotlin.jvm.JvmName("RangesKt")

package kotlin.ranges

/**

 * Represents a range of [Comparable] values.

 */

private open class ComparableRange<T : Comparable<T>>(

 override val start: T,

 override val endInclusive: T

): ClosedRange<T> {

 override fun equals(other: Any?): Boolean {

 return other is ComparableRange<*> && (isEmpty() && other.isEmpty()) ||

 start == other.start && endInclusive == other.endInclusive

 }

 override fun hashCode(): Int {

 return if (isEmpty()) -1 else 31 * start.hashCode() + endInclusive.hashCode()

 }

 override fun toString(): String = "$start..$endInclusive"

}
/**

 * Creates a range from this [Comparable] value to the specified [that] value.

 *

 * This value needs to be smaller than or equal to [that] value, otherwise the returned range will be empty.

 */

@sample samples.ranges.Ranges.rangeFromComparable

public operator fun <T : Comparable<T>> T.rangeTo(that: T): ClosedRange<T> = ComparableRange(this, that)
/**

 * Represents a range of [Comparable] values.

 */

@OptIn(ExperimentalStdlibApi::class)

private open class ComparableOpenEndRange<T : Comparable<T>>(

 override val start: T,

 override val endExclusive: T

): OpenEndRange<T> {

 override fun equals(other: Any?): Boolean {

 return other is ComparableOpenEndRange<*> && (isEmpty() && other.isEmpty()) ||

 start == other.start && endExclusive == other.endExclusive

 }

 override fun hashCode(): Int {

 return if (isEmpty()) -1 else 31 * start.hashCode() + endExclusive.hashCode()

 }

 override fun toString(): String = "$start..<$endExclusive"

}
/**

 * Creates an open-ended range from this [Comparable] value to the specified [that] value.

 *

 * This value needs to be smaller than [that] value, otherwise the returned range will be empty.

 */

@sample samples.ranges.Ranges.rangeFromComparable

public operator fun <T : Comparable<T>> T.rangeUntil(that: T): OpenEndRange<T> = ComparableOpenEndRange(this, that)
/**

 * Represents a range of floating point numbers.

 *

 * Extends [ClosedRange] interface providing custom operation [lessThanOrEquals] for comparing values of range domain type.

 *

 * This interface is implemented by floating point ranges returned by [Float.rangeTo] and [Double.rangeTo] operators to

 * achieve IEEE-754 comparison order instead of total order of floating point numbers.

 */

@SinceKotlin("1.1")

public interface ClosedFloatingPointRange<T : Comparable<T>> : ClosedRange<T> {

 override fun contains(value: T): Boolean = lessThanOrEquals(start, value) && lessThanOrEquals(value, endInclusive)

 override fun isEmpty(): Boolean = !lessThanOrEquals(start, endInclusive)
}
/**

 * Compares two values of range domain type and returns true if first is less than or equal to second.

 */

fun lessThanOrEquals(a: T, b: T): Boolean

}
/**

 * A closed range of values of type `Double`.

 *

 * Numbers are compared with the ends of this range according to IEEE-754.

 */

private class ClosedDoubleRange(

 start: Double,

 endInclusive: Double

): ClosedFloatingPointRange<Double> {

 private val _start = start

 private val _endInclusive = endInclusive

 override val start: Double get() = _start

 override val endInclusive: Double get() = _endInclusive

 override fun lessThanOrEquals(a: Double, b: Double): Boolean = a <= b

 override fun contains(value: Double): Boolean = value >= _start && value <= _endInclusive

 override fun isEmpty(): Boolean = !(_start <= _endInclusive)
 override fun equals(other: Any?): Boolean {

 return other is ClosedDoubleRange && (isEmpty() && other.isEmpty()) ||

 _start == other._start && _endInclusive == other._endInclusive

 }

 override fun hashCode(): Int {

 return if (isEmpty()) -1 else 31 * _start.hashCode() + _endInclusive.hashCode()

 }

 override fun toString(): String = "$_start..$_endInclusive"

}
/**

 * Creates a range from this [Double] value to the specified [that] value.

 *

 * Numbers are compared with the ends of this range according to IEEE-754.

 */

@sample

```

```

samples.ranges.Ranges.rangeFromDouble\n */\n@SinceKotlin("1.1")\npublic operator fun Double.rangeTo(that:
Double): ClosedFloatingPointRange<Double> = ClosedDoubleRange(this, that)\n\n/**\n * An open-ended range of
values of type `Double`.\n * Numbers are compared with the ends of this range according to IEEE-754.\n
*/\n@OptIn(ExperimentalStdlibApi::class)\nprivate class OpenEndDoubleRange(\n start: Double,\n
endExclusive: Double)\n : OpenEndRange<Double> {\n private val _start = start\n private val _endExclusive =
endExclusive\n override val start: Double get() = _start\n override val endExclusive: Double get() =
_endExclusive\n\n private fun lessThanOrEquals(a: Double, b: Double): Boolean = a <= b\n override fun
contains(value: Double): Boolean = value >= _start && value < _endExclusive\n override fun isEmpty(): Boolean
= !(_start < _endExclusive)\n\n override fun equals(other: Any?): Boolean {\n return other is
OpenEndDoubleRange && (isEmpty() && other.isEmpty()) ||\n _start == other._start && _endExclusive
== other._endExclusive)\n }\n\n override fun hashCode(): Int {\n return if (isEmpty()) -1 else 31 *
_start.hashCode() + _endExclusive.hashCode()\n }\n\n override fun toString(): String =
"\$_start..<\$_endExclusive"\n}\n\n/**\n * Creates an open-ended range from this [Double] value to the specified
[that] value.\n * Numbers are compared with the ends of this range according to IEEE-754.\n
*/\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\npublic operator fun Double.rangeUntil(that: Double):
OpenEndRange<Double> = OpenEndDoubleRange(this, that)\n\n/**\n * A closed range of values of type
`Float`.\n * Numbers are compared with the ends of this range according to IEEE-754.\n */\nprivate class
ClosedFloatRange(\n start: Float,\n endInclusive: Float)\n : ClosedFloatingPointRange<Float> {\n private val
_start = start\n private val _endInclusive = endInclusive\n override val start: Float get() = _start\n
override val endInclusive: Float get() = _endInclusive\n\n override fun lessThanOrEquals(a: Float, b: Float): Boolean = a <=
b\n\n override fun contains(value: Float): Boolean = value >= _start && value <= _endInclusive\n
override fun isEmpty(): Boolean = !(_start <= _endInclusive)\n\n override fun equals(other: Any?): Boolean {\n
return other is ClosedFloatRange && (isEmpty() && other.isEmpty()) ||\n _start == other._start &&
_endInclusive == other._endInclusive)\n }\n\n override fun hashCode(): Int {\n return if (isEmpty()) -1 else
31 * _start.hashCode() + _endInclusive.hashCode()\n }\n\n override fun toString(): String =
"\$_start..$_endInclusive"\n}\n\n/**\n * Creates a range from this [Float] value to the specified [that] value.\n *
Numbers are compared with the ends of this range according to IEEE-754.\n * @sample
samples.ranges.Ranges.rangeFromFloat\n */\n@SinceKotlin("1.1")\npublic operator fun Float.rangeTo(that:
Float): ClosedFloatingPointRange<Float> = ClosedFloatRange(this, that)\n\n/**\n * An open-ended range of
values of type `Float`.\n * Numbers are compared with the ends of this range according to IEEE-754.\n
*/\n@OptIn(ExperimentalStdlibApi::class)\nprivate class OpenEndFloatRange(\n start: Float,\n endExclusive:
Float)\n : OpenEndRange<Float> {\n private val _start = start\n private val _endExclusive = endExclusive\n
override val start: Float get() = _start\n override val endExclusive: Float get() = _endExclusive\n\n private fun
lessThanOrEquals(a: Float, b: Float): Boolean = a <= b\n\n override fun contains(value: Float): Boolean = value
>= _start && value < _endExclusive\n\n override fun isEmpty(): Boolean = !(_start < _endExclusive)\n\n
override fun equals(other: Any?): Boolean {\n return other is OpenEndFloatRange && (isEmpty() && other.isEmpty())
||\n _start == other._start && _endExclusive == other._endExclusive)\n }\n\n override fun hashCode():
Int {\n return if (isEmpty()) -1 else 31 * _start.hashCode() + _endExclusive.hashCode()\n }\n\n
override fun toString(): String = "\$_start..<\$_endExclusive"\n}\n\n/**\n * Creates an open-ended range from this [Float] value
to the specified [that] value.\n * Numbers are compared with the ends of this range according to IEEE-754.\n
*/\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\npublic operator fun Float.rangeUntil(that: Float):
OpenEndRange<Float> = OpenEndFloatRange(this, that)\n\n/**\n * Returns `true` if this iterable range contains
the specified [element].\n * Always returns `false` if the [element] is `null`.\n
*/\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline operator fun <T, R> R.contains(element: T?):
Boolean where T : Any, R : ClosedRange<T>, R : Iterable<T> =\n element != null && contains(element)\n\n/**\n
* Returns `true` if this iterable range contains the specified [element].\n * Always returns `false` if the [element]
is `null`.\n */\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline
operator fun <T, R> R.contains(element: T?): Boolean where T : Any, R : OpenEndRange<T>, R : Iterable<T> =\n

```



```

element != null && contains(element)\n\ninternal fun checkStepIsPositive(isPositive: Boolean, step: Number) {\n
if (!isPositive) throw IllegalArgumentException("\nStep must be positive, was: $step.\n")\n}\n"/*\n * Copyright
2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmName("\nKClasses\n")\n@file:Suppress("\nUNCHECKED_CAST\n")\n\npackage
kotlin.reflect\n\nimport kotlin.internal.LowPriorityInOverloadResolution\n\n/**\n * Casts the given [value] to the
class represented by this [KClass] object.\n * Throws an exception if the value is `null` or if it is not an instance of
this class.\n * This is an experimental function that behaves as a similar function from kotlin.reflect.full on
JVM.\n * See [KClass.isInstance]\n * @see [KClass.safeCast]\n
*/\n\n@SinceKotlin("\n1.4\n")\n@WasExperimental(ExperimentalStdlibApi::class)\n@LowPriorityInOverloadResoluti
on\nfun <T : Any> KClass<T>.cast(value: Any?): T {\n if (!isInstance(value)) throw ClassCastException("\nValue
cannot be cast to $qualifiedOrSimpleName\n")\n return value as T\n}\n\n// TODO: replace with qualifiedName
when it is fully supported in K/JS\n\ninternal expect val KClass<*>.qualifiedOrSimpleName: String?\n\n/**\n * Casts
the given [value] to the class represented by this [KClass] object.\n * Returns `null` if the value is `null` or if it is not
an instance of this class.\n * This is an experimental function that behaves as a similar function from
kotlin.reflect.full on JVM.\n * @see [KClass.isInstance]\n * @see [KClass.cast]\n
*/\n\n@SinceKotlin("\n1.4\n")\n@WasExperimental(ExperimentalStdlibApi::class)\n@LowPriorityInOverloadResoluti
on\nfun <T : Any> KClass<T>.safeCast(value: Any?): T? {\n return if (isInstance(value)) value as T else
null\n}\n"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\npackage kotlin.reflect\n\nimport kotlin.jvm.JvmField\nimport kotlin.jvm.JvmStatic\n\n\n/**\n * Represents
a type projection. Type projection is usually the argument to another type in a type usage.\n * For example, in the
type `Array<out Number>`, `out Number` is the covariant projection of the type represented by the class
`Number`.\n * Type projection is either the star projection, or an entity consisting of a specific type plus optional
variance.\n * See the [Kotlin language documentation](https://kotlinlang.org/docs/reference/generics.html#type-
projections)\n * for more information.\n */\n\n@SinceKotlin("\n1.1\n")\npublic data class KTypeProjection
constructor(\n /**\n * The use-site variance specified in the projection, or `null` if this is a star projection.\n
*/\n variance: KVariance?,\n /**\n * The type specified in the projection, or `null` if this is a star
projection.\n */\n type: KType?) {\n init {\n require((variance == null) == (type == null))\n
 if (variance == null)\n "\nStar projection must have no type specified.\n" else\n
"\nThe projection variance $variance requires type to be specified.\n" }\n\n override fun toString():
String = when (variance) {\n null -> "\n*\n" KVariance.INVARIANT -> type.toString()\n
KVariance.IN -> "\nin $type\n" KVariance.OUT -> "\nout $type\n" }\n\n public companion object {\n //
provided for compiler access\n @JvmField\n @PublishedApi\n internal val star: KTypeProjection =
KTypeProjection(null, null)\n /**\n * Star projection, denoted by the `*` character.\n * For example,
in the type `KClass<*>`, `*` is the star projection.\n * See the [Kotlin language
documentation](https://kotlinlang.org/docs/reference/generics.html#star-projections)\n * for more
information.\n */\n public val STAR: KTypeProjection get() = star\n /**\n * Creates an
invariant projection of a given type. Invariant projection is just the type itself,\n * without any use-site variance
modifiers applied to it.\n * For example, in the type `Set<String>`, `String` is an invariant projection of the type
represented by the class `String`.\n */\n @JvmStatic\n public fun invariant(type: KType):
KTypeProjection =\n KTypeProjection(KVariance.INVARIANT, type)\n /**\n * Creates a
contravariant projection of a given type, denoted by the `in` modifier applied to a type.\n * For example, in the
type `MutableList<in Number>`, `in Number` is a contravariant projection of the type of class `Number`.\n */\n
 @JvmStatic\n public fun contravariant(type: KType): KTypeProjection =\n
 KTypeProjection(KVariance.IN, type)\n /**\n * Creates a covariant projection of a given type, denoted
by the `out` modifier applied to a type.\n * For example, in the type `Array<out Number>`, `out Number` is a
covariant projection of the type of class `Number`.\n */\n @JvmStatic\n public fun covariant(type:

```

```

KType): KTypeProjection =\n KTypeProjection(KVariance.OUT, type)\n }\n}", /*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\npackage kotlin.reflect\n\n/**\n * Represents variance applied to a type parameter on the declaration site (*declaration-site variance*),\n * or to a type in a projection (*use-site variance*).\n *\n * See the [Kotlin language documentation](https://kotlinlang.org/docs/reference/generics.html#variance)\n * for more information.\n *\n * @see [KTypeParameter.variance]\n * @see [KTypeProjection]\n *\n@SinceKotlin("1.1")\nenum class KVariance {\n /**\n * The affected type parameter or type is *invariant*, which means it has no variance applied to it.\n *\n * INVARIANT,\n }\n /**\n * The affected type parameter or type is *contravariant*. Denoted by the `in` modifier in the source code.\n *\n * IN,\n }\n /**\n * The affected type parameter or type is *covariant*. Denoted by the `out` modifier in the source code.\n *\n * OUT,\n }\n}", /*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\npackage kotlin.reflect\n\n/**\n * Returns a runtime representation of the given reified type [T] as an instance of [KType].\n *\n * Note that on JVM, the created type has no annotations ([KType.annotations] returns an empty list)\n * even if the type in the source code is annotated. Support for type annotations might be added in a future version.\n *\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <reified T> typeOf(): KType =\n throw UnsupportedOperationException("This function is implemented as an intrinsic on all supported platforms.")\n}", /*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\npackage kotlin.jvm\n\npackage kotlin.text\n\n/**\n * An object to which char sequences and values can be appended.\n *\nexpect interface Appendable {\n /**\n * Appends the specified character [value] to this Appendable and returns this instance.\n *\n * @param value the character to append.\n *\n fun append(value: Char): Appendable\n\n /**\n * Appends the specified character sequence [value] to this Appendable and returns this instance.\n *\n * @param value the character sequence to append. If [value] is `null`, then the four characters `\\`null` are appended to this Appendable.\n *\n fun append(value: CharSequence?): Appendable\n\n /**\n * Appends a subsequence of the specified character sequence [value] to this Appendable and returns this instance.\n *\n * @param value the character sequence from which a subsequence is appended. If [value] is `null`,\n * then characters are appended as if [value] contained the four characters `\\`null`. \n * @param startIndex the beginning (inclusive) of the subsequence to append.\n * @param endIndex the end (exclusive) of the subsequence to append.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`. \n *\n fun append(value: CharSequence?, startIndex: Int, endIndex: Int): Appendable\n}\n\n/**\n * Appends a subsequence of the specified character sequence [value] to this Appendable and returns this instance.\n *\n * @param value the character sequence from which a subsequence is appended.\n * @param startIndex the beginning (inclusive) of the subsequence to append.\n * @param endIndex the end (exclusive) of the subsequence to append.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`. \n *\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T : Appendable> T.appendRange(value: CharSequence, startIndex: Int, endIndex: Int): T {\n @Suppress("UNCHECKED_CAST")\n return append(value, startIndex, endIndex) as T\n}\n\n/**\n * Appends all arguments to the given [Appendable].\n *\npublic fun <T : Appendable> T.append(vararg value: CharSequence?): T {\n for (item in value)\n append(item)\n return this\n}\n\n/**\n * Appends a line feed character (`\\`n`) to this Appendable.\n *\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun Appendable.appendLine(): Appendable = append("\\`n")\n\n/**\n * Appends value to the given Appendable and a line feed character (`\\`n`) after it.\n *\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun

```



```

license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/**\n * Defines names for Unicode symbols used in proper
Typography.\n */\npublic object Typography {\n /** The character " \u2013 quotation mark */\n public
const val quote: Char = "\u0022"\n /** The character $ \u2013 dollar sign */\n public const val dollar:
Char = "\u0024"\n /** The character & \u2013 ampersand */\n public const val amp: Char = "\u0026"\n
 /** The character < \u2013 less-than sign */\n public const val less: Char = "\u003C"\n /** The character
> \u2013 greater-than sign */\n public const val greater: Char = "\u003E"\n /** The non-breaking space
character */\n public const val nbsp: Char = "\u00A0"\n /** The character × */\n public const val times:
Char = "\u00D7"\n /** The character ¢ */\n public const val cent: Char = "\u00A2"\n /** The character
£ */\n public const val pound: Char = "\u00A3"\n /** The character § */\n public const val
section: Char = "\u00A7"\n /** The character © */\n public const val copyright: Char = "\u00A9"\n /**
The character « */\n @SinceKotlin("1.6")\n public const val leftGuillemet: Char = "\u00AB"\n /**
The character » */\n @SinceKotlin("1.6")\n public const val rightGuillemet: Char = "\u00BB"\n /**
The character ® */\n public const val registered: Char = "\u00AE"\n /** The character ° */\n
 public const val degree: Char = "\u00B0"\n /** The character ± */\n public const val plusMinus: Char =
"\u00B1"\n /** The character ¶ */\n public const val paragraph: Char = "\u00B6"\n /** The character
· */\n public const val middleDot: Char = "\u00B7"\n /** The character ½ */\n public const val
half: Char = "\u00BD"\n /** The character – */\n public const val ndash: Char = "\u2013"\n /** The
character — */\n public const val mdash: Char = "\u2014"\n /** The character ‘ */\n public
const val leftSingleQuote: Char = "\u2018"\n /** The character ’ */\n public const val rightSingleQuote:
Char = "\u2019"\n /** The character ‚ */\n public const val lowSingleQuote: Char = "\u201A"\n /**
The character “ */\n public const val leftDoubleQuote: Char = "\u201C"\n /** The character ”
*/\n public const val rightDoubleQuote: Char = "\u201D"\n /** The character „ */\n public const val
lowDoubleQuote: Char = "\u201E"\n /** The character † */\n public const val dagger: Char =
"\u2020"\n /** The character ‡ */\n public const val doubleDagger: Char = "\u2021"\n /** The
character • */\n public const val bullet: Char = "\u2022"\n /** The character … */\n public
const val ellipsis: Char = "\u2026"\n /** The character ′ */\n public const val prime: Char = "\u2032"\n
 /** The character ″ */\n public const val doublePrime: Char = "\u2033"\n /** The character €
*/\n public const val euro: Char = "\u20AC"\n /** The character ™ */\n public const val tm: Char =
"\u2122"\n /** The character ≈ */\n public const val almostEqual: Char = "\u2248"\n /** The character
≠ */\n public const val notEqual: Char = "\u2260"\n /** The character ≤ */\n public const val
lessOrEqual: Char = "\u2264"\n /** The character ≥ */\n public const val greaterOrEqual: Char =
"\u2265"\n\n /** The character « */\n @Deprecated("This constant has a typo in the name. Use
leftGuillemet instead.", ReplaceWith("Typography.leftGuillemet"))\n @DeprecatedSinceKotlin("1.6")\n public const val leftGuillemete: Char = "\u00AB"\n\n /** The character » */\n @Deprecated("This
constant has a typo in the name. Use rightGuillemet instead.", ReplaceWith("Typography.rightGuillemet"))\n @DeprecatedSinceKotlin("1.6")\n public const val rightGuillemete: Char = "\u00BB"\n}\n\n * Copyright
2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/**\n * Represents a collection of captured groups in a single match of a regular expression.\n */\n * This collection has size
of `groupCount + 1` where `groupCount` is the count of groups in the regular expression.\n * Groups are indexed
from 1 to `groupCount` and group with the index 0 corresponds to the entire match.\n */\n * An element of the
collection at the particular index can be `null` if the corresponding group in the regular expression is optional
and\n * there was no match captured by that group.\n */\npublic interface MatchGroupCollection :
Collection<MatchGroup?> {\n /** Returns a group with the specified [index].\n */\n * @return An instance
of [MatchGroup] if the group with the specified [index] was matched or `null` otherwise.\n */\n * Groups are
indexed from 1 to the count of groups in the regular expression. A group with the index 0\n * corresponds to the
entire match.\n */\n public operator fun get(index: Int): MatchGroup?\n}\n\n * Extends
[MatchGroupCollection] by introducing a way to get matched groups by name, when regex supports it.\n

```

```

*^@SinceKotlin("1.1")\npublic interface MatchNamedGroupCollection : MatchGroupCollection {\n /**\n *
Returns a named group with the specified [name].\n * @return An instance of [MatchGroup] if the group with the
specified [name] was matched or `null` otherwise.\n * @throws IllegalArgumentException if there is no group
with the specified [name] defined in the regex pattern.\n * @throws UnsupportedOperationException if this
match group collection doesn't support getting match groups by name,\n * for example, when it's not supported by
the current platform.\n */\n public operator fun get(name: String): MatchGroup?\n}\n\n/**\n * Represents the
results from a single regular expression match.\n */\npublic interface MatchResult {\n /** The range of indices in
the original string where match was captured. */\n public val range: IntRange\n /** The substring from the input
string captured by this match. */\n public val value: String\n /**\n * A collection of groups matched by the
regular expression.\n *\n * This collection has size of `groupCount + 1` where `groupCount` is the count of
groups in the regular expression.\n *\n * Groups are indexed from 1 to `groupCount` and group with the index 0
corresponds to the entire match.\n */\n public val groups: MatchGroupCollection\n /**\n * A list of
matched indexed group values.\n *\n * This list has size of `groupCount + 1` where `groupCount` is the count
of groups in the regular expression.\n *\n * Groups are indexed from 1 to `groupCount` and group with the index 0
corresponds to the entire match.\n *\n * If the group in the regular expression is optional and there were no
match captured by that group,\n * corresponding item in [groupValues] is an empty string.\n *\n * @sample
samples.text.Regexps.matchDestructuringToGroupValues\n */\n public val groupValues: List<String>\n}\n\n/**\n * An instance of [MatchResult.Destructured] wrapper providing components for destructuring assignment
of group values.\n *\n * component1 corresponds to the value of the first group, component2 \u2014 of the
second, and so on.\n *\n * @sample samples.text.Regexps.matchDestructuringToGroupValues\n */\n
public val destructured: Destructured get() = Destructured(this)\n\n /** Returns a new [MatchResult] with the
results for the next match, starting at the position\n * at which the last match ended (at the character after the last
matched character).\n */\n public fun next(): MatchResult?\n\n /**\n * Provides components for
destructuring assignment of group values.\n *\n * [component1] corresponds to the value of the first group,
[component2] \u2014 of the second, and so on.\n *\n * If the group in the regular expression is optional and
there were no match captured by that group,\n * corresponding component value is an empty string.\n *\n *
@sample samples.text.Regexps.matchDestructuringToGroupValues\n */\n public class Destructured internal
constructor(public val match: MatchResult) {\n @kotlin.internal.InlineOnly\n public operator inline fun
component1(): String = match.groupValues[1]\n @kotlin.internal.InlineOnly\n public operator inline fun
component2(): String = match.groupValues[2]\n @kotlin.internal.InlineOnly\n public operator inline fun
component3(): String = match.groupValues[3]\n @kotlin.internal.InlineOnly\n public operator inline fun
component4(): String = match.groupValues[4]\n @kotlin.internal.InlineOnly\n public operator inline fun
component5(): String = match.groupValues[5]\n @kotlin.internal.InlineOnly\n public operator inline fun
component6(): String = match.groupValues[6]\n @kotlin.internal.InlineOnly\n public operator inline fun
component7(): String = match.groupValues[7]\n @kotlin.internal.InlineOnly\n public operator inline fun
component8(): String = match.groupValues[8]\n @kotlin.internal.InlineOnly\n public operator inline fun
component9(): String = match.groupValues[9]\n @kotlin.internal.InlineOnly\n public operator inline fun
component10(): String = match.groupValues[10]\n\n /**\n * Returns destructured group values as a list of
strings.\n * First value in the returned list corresponds to the value of the first group, and so on.\n *\n
* @sample samples.text.Regexps.matchDestructuringToGroupValues\n */\n public fun toList():
List<String> = match.groupValues.subList(1, match.groupValues.size)\n }\n}\n\n /**\n * Copyright 2010-2021
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass()\n@file:kotlin.jvm.JvmName("DurationUnitKt")\n\npackage
kotlin.time\n\n/**\n * The list of possible time measurement units, in which a duration can be expressed.\n *\n
* The smallest time unit is [NANOSECONDS] and the largest is [DAYS], which corresponds to exactly 24
[HOURS].\n */\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalTime::class)\npublic expect enum class
DurationUnit {\n /**\n * Time unit representing one nanosecond, which is 1/1000 of a microsecond.\n *\n

```

```

NANOSECONDS, \n /** \n * Time unit representing one microsecond, which is 1/1000 of a millisecond. \n
* \n MICROSECONDS, \n /** \n * Time unit representing one millisecond, which is 1/1000 of a second. \n
* \n MILLISECONDS, \n /** \n * Time unit representing one second. \n * \n SECONDS, \n /** \n *
Time unit representing one minute. \n * \n MINUTES, \n /** \n * Time unit representing one hour. \n * \n
HOURS, \n /** \n * Time unit representing one day, which is always equal to 24 hours. \n * \n
DAYS; \n} \n \n /** Converts the given time duration [value] expressed in the specified [sourceUnit] into the specified
[targetUnit]. * \n @SinceKotlin("1.3") \n internal expect fun convertDurationUnit(value: Double, sourceUnit:
DurationUnit, targetUnit: DurationUnit): Double \n \n // overflown result is
unspecified \n @SinceKotlin("1.5") \n internal expect fun convertDurationUnitOverflow(value: Long, sourceUnit:
DurationUnit, targetUnit: DurationUnit): Long \n \n // overflown result is coerced in the Long range
boundaries \n @SinceKotlin("1.5") \n internal expect fun convertDurationUnit(value: Long, sourceUnit:
DurationUnit, targetUnit: DurationUnit):
Long \n \n @SinceKotlin("1.3") \n @Suppress("REDUNDANT_ELSE_IN_WHEN") \n internal fun
DurationUnit.shortName(): String = when (this) { \n DurationUnit.NANOSECONDS -> "ns" \n
DurationUnit.MICROSECONDS -> "us" \n DurationUnit.MILLISECONDS -> "ms" \n
DurationUnit.SECONDS -> "s" \n DurationUnit.MINUTES -> "m" \n DurationUnit.HOURS -> "h" \n
DurationUnit.DAYS -> "d" \n else -> error("Unknown unit: $this") \n} \n \n @SinceKotlin("1.5") \n internal fun
durationUnitByShortName(shortName: String): DurationUnit = when (shortName) { \n "ns" ->
DurationUnit.NANOSECONDS \n "us" -> DurationUnit.MICROSECONDS \n "ms" ->
DurationUnit.MILLISECONDS \n "s" -> DurationUnit.SECONDS \n "m" -> DurationUnit.MINUTES \n
"h" -> DurationUnit.HOURS \n "d" -> DurationUnit.DAYS \n else -> throw
IllegalArgumentExcepTion("Unknown duration unit short name:
$shortName") \n} \n \n @SinceKotlin("1.5") \n internal fun durationUnitByIsoChar(isoChar: Char,
isTimeComponent: Boolean): DurationUnit = \n when { \n !isTimeComponent -> { \n when (isoChar)
{ \n 'D' -> DurationUnit.DAYS \n else -> throw IllegalArgumentExcepTion("Invalid or
unsupported duration ISO non-time unit: $isoChar") \n } \n } \n else -> { \n when (isoChar) { \n
'H' -> DurationUnit.HOURS \n 'M' -> DurationUnit.MINUTES \n 'S' ->
DurationUnit.SECONDS \n else -> throw IllegalArgumentExcepTion("Invalid duration ISO time unit:
$isoChar") \n } \n } \n }, "/* \n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming
Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file. \n * \n \n package kotlin.time \n \n import kotlin.annotation.AnnotationTarget.* \n \n /** \n *
This annotation marks the experimental preview of the standard library API for measuring time and working with
durations. \n * \n * > Note that this API is in a preview state and has a very high chance of being changed in the
future. \n * Do not use it if you develop a library since your library will become binary incompatible \n * with the
future versions of the standard library. \n * \n * Any usage of a declaration annotated with `@ExperimentalTime`
must be accepted either by \n * annotating that usage with the [OptIn] annotation, e.g.
`@OptIn(ExperimentalTime::class)`, \n * or by using the compiler argument ` -opt-
in=kotlin.time.ExperimentalTime`. \n * \n @RequiresOptIn(level =
RequiresOptIn.Level.ERROR) \n @MustBeDocumented \n @Retention(AnnotationRetention.BINARY) \n @Target(\n
CLASS, \n ANNOTATION_CLASS, \n PROPERTY, \n FIELD, \n LOCAL_VARIABLE, \n
VALUE_PARAMETER, \n CONSTRUCTOR, \n FUNCTION, \n PROPERTY_GETTER, \n
PROPERTY_SETTER, \n TYPEALIAS) \n @SinceKotlin("1.3") \n public annotation class
ExperimentalTime \n "/* \n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language
contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file. \n * \n \n package kotlin.time \n \n import kotlin.jvm.JvmInline \n \n /** \n * A source of time
for measuring time intervals. \n * \n * The only operation provided by the time source is [markNow]. It returns a
[TimeMark], which can be used to query the elapsed time later. \n * \n * @see [measureTime] \n * @see
[measureTimedValue] \n * \n @SinceKotlin("1.3") \n @ExperimentalTime \n public interface TimeSource { \n /** \n

```

```

 * Marks a point in time on this time source.\n
 * The returned [TimeMark] instance encapsulates the captured time point and allows querying\n
 * the duration of time interval [elapsed][TimeMark.elapsedNow] from that point.\n
 * public fun markNow(): TimeMark\n
 * The most precise time source available in the platform.\n
 * This time source returns its readings from a source of monotonic time when it is available in a target platform,\n
 * and resorts to a non-monotonic time source otherwise.\n
 * The function [markNow] of this time source returns the specialized [ValueTimeMark] that is an inline value class\n
 * wrapping a platform-dependent time reading value.\n
 * public object Monotonic : TimeSource {\n
 * override fun markNow(): ValueTimeMark = MonotonicTimeSource.markNow()\n
 * override fun toString(): String = MonotonicTimeSource.toString()\n
 * /**\n
 * * A specialized [kotlin.time.TimeMark] returned by [TimeSource.Monotonic].\n
 * * This time mark is implemented as an inline value class wrapping a platform-dependent\n
 * * time reading value of the default monotonic time source, thus allowing to avoid additional boxing\n
 * * of that value.\n
 * * The operations [plus] and [minus] are also specialized to return [ValueTimeMark] type.\n
 * * @ExperimentalTime\n
 * * @SinceKotlin("1.7")\n
 * * @JvmInline\n
 * * public value class ValueTimeMark internal constructor(internal val reading: ValueTimeMarkReading) : TimeMark {\n
 * * override fun elapsedNow(): Duration = MonotonicTimeSource.elapsedFrom(this)\n
 * * override fun plus(duration: Duration): ValueTimeMark = MonotonicTimeSource.adjustReading(this, duration)\n
 * * override fun minus(duration: Duration): ValueTimeMark = MonotonicTimeSource.adjustReading(this, -duration)\n
 * * override fun hasPassedNow(): Boolean = !elapsedNow().isNegative()\n
 * * override fun hasNotPassedNow(): Boolean = elapsedNow().isNegative()\n
 * * }\n
 * * }\n
 * * public companion object {\n
 * * }\n
 * * }\n
 * * /** A platform-specific reading type that is wrapped by [TimeSource.Monotonic.ValueTimeMark] inline class. *\n
 * * internal expect class ValueTimeMarkReading\n
 * * /**\n
 * * Represents a time point notched on a particular [TimeSource]. Remains bound to the time source it was taken from\n
 * * and allows querying for the duration of time elapsed from that point (see the function [elapsedNow]).\n
 * * @SinceKotlin("1.3")\n
 * * @ExperimentalTime\n
 * * public interface TimeMark {\n
 * * /**\n
 * * Returns the amount of time passed from this mark measured with the time source from which this mark was taken.\n
 * * Note that the value returned by this function can change on subsequent invocations.\n
 * * @throws IllegalArgumentException an implementation may throw if calculating the elapsed time involves\n
 * * adding a positive infinite duration to an infinitely distant past time mark or\n
 * * a negative infinite duration to an infinitely distant future time mark.\n
 * * public abstract fun elapsedNow(): Duration\n
 * * /**\n
 * * Returns a time mark on the same time source that is ahead of this time mark by the specified [duration].\n
 * * The returned time mark is more _late_ when the [duration] is positive, and more _early_ when the [duration] is negative.\n
 * * If the time mark is adjusted too far in the past or in the future, it may saturate to an infinitely distant time mark.\n
 * * In that case, [elapsedNow] will return an infinite duration elapsed from such infinitely distant mark.\n
 * * @throws IllegalArgumentException an implementation may throw if a positive infinite duration is added to an infinitely distant past time mark or\n
 * * a negative infinite duration is added to an infinitely distant future time mark.\n
 * * public open operator fun plus(duration: Duration): TimeMark = AdjustedTimeMark(this, duration)\n
 * * /**\n
 * * Returns a time mark on the same time source that is behind this time mark by the specified [duration].\n
 * * The returned time mark is more _early_ when the [duration] is positive, and more _late_ when the [duration] is negative.\n
 * * If the time mark is adjusted too far in the past or in the future, it may saturate to an infinitely distant time mark.\n
 * * In that case, [elapsedNow] will return an infinite duration elapsed from such infinitely distant mark.\n
 * * @throws IllegalArgumentException an implementation may throw if a positive infinite duration is subtracted from an infinitely distant future time mark or\n
 * * a negative infinite duration is subtracted from an infinitely distant past time mark.\n
 * * public open operator fun minus(duration: Duration): TimeMark = plus(-duration)\n
 * * /**\n
 * * Returns true if this time mark has passed according to the time source from which this mark was taken.\n
 * * Note that the value returned by this function can change on subsequent invocations.\n
 * * If the time source is monotonic, it can change only from `false` to `true`, namely, when the time mark becomes behind the current point of the time source.\n
 * * public fun hasPassedNow(): Boolean = !elapsedNow().isNegative()\n
 * * /**\n
 * * Returns false if this time mark

```

```

has not passed according to the time source from which this mark was taken.\n *\n * Note that the value
returned by this function can change on subsequent invocations.\n * If the time source is monotonic, it can change
only from `true` to `false`, namely, when the time mark becomes behind the current point of the time source.\n
*\n public fun hasNotPassedNow(): Boolean =
elapsedNow().isNegative()\n}\n\n@ExperimentalTime\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@
Deprecated(\n "Subtracting one TimeMark from another is not a well defined operation because these time marks
could have been obtained from the different time sources.",\n level =
DeprecationLevel.ERROR)\n\n@Suppress("UNUSED_PARAMETER")\npublic inline operator fun
TimeMark.minus(other: TimeMark): Duration = throw Error("Operation is
disallowed.")\n\n@ExperimentalTime\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Deprecated(\n
"Comparing one TimeMark to another is not a well defined operation because these time marks could have been
obtained from the different time sources.",\n level =
DeprecationLevel.ERROR)\n\n@Suppress("UNUSED_PARAMETER")\npublic inline operator fun
TimeMark.compareTo(other: TimeMark): Int = throw Error("Operation is
disallowed.")\n\n@ExperimentalTime\nprivate class AdjustedTimeMark(val mark: TimeMark, val adjustment:
Duration) : TimeMark {\n override fun elapsedNow(): Duration = mark.elapsedNow() - adjustment\n override
fun plus(duration: Duration): TimeMark = AdjustedTimeMark(mark, adjustment + duration)\n}\n"/*\n *
Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.time\n\n@SinceKotlin("1.3")\n@ExperimentalTime\ninternal expect object MonotonicTimeSource :
TimeSource {\n override fun markNow(): TimeSource.Monotonic.ValueTimeMark\n fun
elapsedFrom(timeMark: TimeSource.Monotonic.ValueTimeMark): Duration\n fun adjustReading(timeMark:
TimeSource.Monotonic.ValueTimeMark, duration: Duration): TimeSource.Monotonic.ValueTimeMark\n}\n\n/**\n *
An abstract class used to implement time sources that return their readings as [Long] values in the specified
[unit].\n * @property unit The unit in which this time source's readings are expressed.\n
*/\n\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic abstract class AbstractLongTimeSource(protected val
unit: DurationUnit) : TimeSource {\n /**\n * This protected method should be overridden to return the current
reading of the time source expressed as a [Long] number\n * in the unit specified by the [unit] property.\n
*/\n\n protected abstract fun read(): Long\n\n private class LongTimeMark(private val startedAt: Long, private val
timeSource: AbstractLongTimeSource, private val offset: Duration) : TimeMark {\n override fun elapsedNow():
Duration = (timeSource.read() - startedAt).toDuration(timeSource.unit) - offset\n override fun plus(duration:
Duration): TimeMark = LongTimeMark(startedAt, timeSource, offset + duration)\n }\n\n override fun
markNow(): TimeMark = LongTimeMark(read(), this, Duration.ZERO)\n}\n\n/**\n * An abstract class used to
implement time sources that return their readings as [Double] values in the specified [unit].\n * @property unit
The unit in which this time source's readings are expressed.\n
*/\n\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic abstract class AbstractDoubleTimeSource(protected val
unit: DurationUnit) : TimeSource {\n /**\n * This protected method should be overridden to return the current
reading of the time source expressed as a [Double] number\n * in the unit specified by the [unit] property.\n
*/\n\n protected abstract fun read(): Double\n\n private class DoubleTimeMark(private val startedAt: Double,
private val timeSource: AbstractDoubleTimeSource, private val offset: Duration) : TimeMark {\n override fun
elapsedNow(): Duration = (timeSource.read() - startedAt).toDuration(timeSource.unit) - offset\n override fun
plus(duration: Duration): TimeMark = DoubleTimeMark(startedAt, timeSource, offset + duration)\n }\n\n override fun
markNow(): TimeMark = DoubleTimeMark(read(), this, Duration.ZERO)\n}\n\n/**\n * A time source
that has programmatically updatable readings. It is useful as a predictable source of time in tests.\n * @property
unit The unit in which this time source's readings are expressed.\n * @property duration The current
reading value can be advanced by the specified duration amount with the operator [plusAssign]:\n * val
timeSource = TestTimeSource()\n * timeSource += 10.seconds\n * Implementation note: the current
reading value is stored as a [Long] number of nanoseconds,\n * thus it's capable to represent a time range of
approximately \u00b11292 years.\n * Should the reading value overflow as the result of [plusAssign] operation, an

```



```

[IllegalStateException] is thrown.\n *\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic class TestTimeSource
: AbstractLongTimeSource(unit = DurationUnit.NANOSECONDS) {\n private var reading: Long = 0L\n\n override fun read(): Long = reading\n\n /**\n * Advances the current reading value of this time source by the
specified [duration].\n *\n * [duration] value is rounded down towards zero when converting it to a [Long]
number of nanoseconds.\n * For example, if the duration being added is `0.6.nanoseconds`, the reading doesn't
advance because\n * the duration value is rounded to zero nanoseconds.\n *\n * @throws
IllegalStateException when the reading value overflows as the result of this operation.\n *\n public operator fun
plusAssign(duration: Duration) {\n val longDelta = duration.toLong(unit)\n reading = if (longDelta !=
Long.MIN_VALUE && longDelta != Long.MAX_VALUE) {\n // when delta fits in long, add it as long\n
 val newReading = reading + longDelta\n if (reading xor longDelta >= 0 && reading xor newReading < 0)
overflow(duration)\n newReading\n } else {\n val delta = duration.toDouble(unit)\n // when
delta is greater than long, add it as double\n val newReading = reading + delta\n if (newReading >
Long.MAX_VALUE || newReading < Long.MIN_VALUE) overflow(duration)\n newReading.toLong()\n }\n }\n\n private fun overflow(duration: Duration) {\n throw IllegalStateException("TestTimeSource will
overflow if its reading ${reading}ns is advanced by $duration.")\n }\n }\n\n /**\n * Copyright 2010-2022 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n *\n package kotlin.time\n import
kotlin.time.Duration.Companion.milliseconds\n import kotlin.time.Duration.Companion.nanoseconds\n\n // Long
time reading saturation math, shared between JVM and Native\n\n internal fun saturatingAdd(longNs: Long,
duration: Duration): Long {\n val durationNs = duration.inWholeNanoseconds\n if ((longNs - 1) or 1 ==
Long.MAX_VALUE) { // MIN_VALUE or MAX_VALUE - the reading is infinite\n return
checkInfiniteSumDefined(longNs, duration, durationNs)\n }\n if ((durationNs - 1) or 1 == Long.MAX_VALUE)
{ // duration doesn't fit in Long nanos\n return saturatingAddInHalves(longNs, duration)\n }\n val result =
longNs + durationNs\n if (((longNs xor result) and (durationNs xor result)) < 0) {\n return if (longNs < 0)
Long.MIN_VALUE else Long.MAX_VALUE\n }\n return result\n }\n\n private fun
checkInfiniteSumDefined(longNs: Long, duration: Duration, durationNs: Long): Long {\n if (duration.isInfinite()
&& (longNs xor durationNs < 0)) throw IllegalArgumentException("Summing infinities of different signs")\n return
longNs\n }\n\n private fun saturatingAddInHalves(longNs: Long, duration: Duration): Long {\n val half =
duration / 2\n if ((half.inWholeNanoseconds - 1) or 1 == Long.MAX_VALUE) {\n // this will definitely
saturate\n return (longNs + duration.toDouble(DurationUnit.NANOSECONDS)).toLong()\n } else {\n return
saturatingAdd(saturatingAdd(longNs, half), half)\n }\n }\n\n internal fun saturatingDiff(valueNs: Long,
originNs: Long): Duration {\n if ((originNs - 1) or 1 == Long.MAX_VALUE) { // MIN_VALUE or
MAX_VALUE\n return -(originNs.toDuration(DurationUnit.DAYS)) // saturate to infinity\n }\n val result =
valueNs - originNs\n if ((result xor valueNs) and (result xor originNs).inv() < 0) {\n val resultMs = valueNs /
NANOS_IN_MILLIS - originNs / NANOS_IN_MILLIS\n val resultNs = valueNs % NANOS_IN_MILLIS -
originNs % NANOS_IN_MILLIS\n return resultMs.milliseconds + resultNs.nanoseconds\n }\n return
result.nanoseconds\n }\n\n /**\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n *\n package kotlin.time\n\n import kotlin.contracts.*\n\n /**\n * Executes the given
function [block] and returns the duration of elapsed time interval.\n *\n * The elapsed time is measured with
[TimeSource.Monotonic].\n *\n @SinceKotlin("1.3")\n @ExperimentalTime\n public inline fun
measureTime(block: () -> Unit): Duration {\n contract {\n callsInPlace(block,
InvocationKind.EXACTLY_ONCE)\n }\n return TimeSource.Monotonic.measureTime(block)\n }\n\n /**\n * Executes the given function [block] and returns the duration of elapsed time interval.\n *\n * The elapsed time is
measured with the specified `this` [TimeSource] instance.\n *\n @SinceKotlin("1.3")\n @ExperimentalTime\n public inline fun TimeSource.measureTime(block: () -> Unit):
Duration {\n contract {\n callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n }\n val mark =
markNow()\n block()\n return mark.elapsedNow()\n }\n\n /**\n * Executes the given function [block] and returns

```

```

the duration of elapsed time interval.\n * The elapsed time is measured with the specified `this`
[TimeSource.Monotonic] instance.\n *^@SinceKotlin("1.7")\n@ExperimentalTime\npublic inline fun
TimeSource.Monotonic.measureTime(block: () -> Unit): Duration {\n contract {\n callsInPlace(block,
InvocationKind.EXACTLY_ONCE)\n }\n\n val mark = markNow()\n block()\n return
mark.elapsedNow()\n}\n\n/**\n * Data class representing a result of executing an action, along with the duration
of elapsed time interval.\n * \n * @property value the result of the action.\n * @property duration the time elapsed to
execute the action.\n *^@SinceKotlin("1.3")\n@ExperimentalTime\npublic data class TimedValue<T>(val
value: T, val duration: Duration)\n\n/**\n * Executes the given function [block] and returns an instance of
[TimedValue] class, containing both\n * the result of the function execution and the duration of elapsed time
interval.\n * \n * The elapsed time is measured with [TimeSource.Monotonic].\n
*^@SinceKotlin("1.3")\n@ExperimentalTime\npublic inline fun <T> measureTimedValue(block: () -> T):
TimedValue<T> {\n contract {\n callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n }\n\n return
TimeSource.Monotonic.measureTimedValue(block)\n}\n\n/**\n * Executes the given [block] and returns an
instance of [TimedValue] class, containing both\n * the result of function execution and the duration of elapsed time
interval.\n * \n * The elapsed time is measured with the specified `this` [TimeSource] instance.\n
*^@SinceKotlin("1.3")\n@ExperimentalTime\npublic inline fun <T> TimeSource.measureTimedValue(block: ()
-> T): TimedValue<T> {\n contract {\n callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n }\n\n val
mark = markNow()\n val result = block()\n return TimedValue(result, mark.elapsedNow())\n}\n\n/**\n *
Executes the given [block] and returns an instance of [TimedValue] class, containing both\n * the result of function
execution and the duration of elapsed time interval.\n * \n * The elapsed time is measured with the specified `this`
[TimeSource.Monotonic] instance.\n *^@SinceKotlin("1.7")\n@ExperimentalTime\npublic inline fun <T>
TimeSource.Monotonic.measureTimedValue(block: () -> T): TimedValue<T> {\n contract {\n
callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n }\n\n val mark = markNow()\n val result =
block()\n return TimedValue(result, mark.elapsedNow())\n}\n\n"/**\n * Copyright 2010-2020 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n *^@n\npackage kotlin\n\nimport kotlin.coroutines.*\nimport
kotlin.coroutines.intrinsics.*\nimport kotlin.native.concurrent.SharedImmutable\n\n/**\n * Defines deep recursive
function that keeps its stack on the heap,\n * which allows very deep recursive computations that do not use the
actual call stack.\n * To initiate a call to this deep recursive function use its [invoke] function.\n * As a rule of
thumb, it should be used if recursion goes deeper than a thousand calls.\n * \n * The [DeepRecursiveFunction] takes
one parameter of type [T] and returns a result of type [R].\n * \n * The [block] of code defines the body of a recursive
function. In this block\n * [callRecursive][DeepRecursiveScope.callRecursive] function can be used to make a
recursive call\n * to the declared function. Other instances of [DeepRecursiveFunction] can be called\n * in this
scope with `callRecursive` extension, too.\n * \n * For example, take a look at the following recursive tree class and
a deeply\n * recursive instance of this tree with 100K nodes:\n * \n * ```\n * class Tree(val left: Tree? = null, val
right: Tree? = null)\n * val deepTree = generateSequence(Tree()) { Tree(it) }.take(100_000).last()\n * ```\n * \n * A
regular recursive function can be defined to compute a depth of a tree:\n * \n * ```\n * fun depth(t: Tree?): Int =\n *
if (t == null) 0 else max(depth(t.left), depth(t.right)) + 1\n * println(depth(deepTree)) // StackOverflowError\n * ```\n *
\n * If this `depth` function is called for a `deepTree` it produces `StackOverflowError` because of deep
recursion.\n * However, the `depth` function can be rewritten using `DeepRecursiveFunction` in the following way,
and then\n * it successfully computes [depth(deepTree)][DeepRecursiveFunction.invoke] expression:\n * \n * ```\n
* val depth = DeepRecursiveFunction<Tree?, Int> { t ->\n * if (t == null) 0 else max(callRecursive(t.left),
callRecursive(t.right)) + 1\n * }\n * println(depth(deepTree)) // Ok\n * ```\n * \n * Deep recursive functions can also
mutually call each other using a heap for the stack via\n * [callRecursive][DeepRecursiveScope.callRecursive]
extension. For example, the\n * following pair of mutually recursive functions computes the number of tree nodes at
even depth in the tree.\n * \n * ```\n * val mutualRecursion = object {\n * val even:
DeepRecursiveFunction<Tree?, Int> = DeepRecursiveFunction { t ->\n * if (t == null) 0 else
odd.callRecursive(t.left) + odd.callRecursive(t.right) + 1\n * }\n * val odd: DeepRecursiveFunction<Tree?,

```

```

Int> = DeepRecursiveFunction { t ->\n * if (t == null) 0 else even.callRecursive(t.left) +
even.callRecursive(t.right)\n * }\n * }\n * ```\n * }\n * @param [T] the function parameter type.\n * @param [R]
the function result type.\n * @param block the function body.\n
*\n@SinceKotlin("1.7")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic class
DeepRecursiveFunction<T, R>(\n internal val block: suspend DeepRecursiveScope<T, R>.(T) -> R)\n)\n\n/**\n *
Initiates a call to this deep recursive function, forming a root of the call tree.\n * This operator should not be
used from inside of [DeepRecursiveScope] as it uses the call stack slot for\n * initial recursive invocation. From
inside of [DeepRecursiveScope] use\n * [callRecursive][DeepRecursiveScope.callRecursive].\n
*\n@SinceKotlin("1.7")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic operator fun <T, R>
DeepRecursiveFunction<T, R>.invoke(value: T): R =\n DeepRecursiveScopeImpl<T, R>(block,
value).runCallLoop()\n)\n\n/**\n * A scope class for [DeepRecursiveFunction] function declaration that defines
[callRecursive] methods to\n * recursively call this function or another [DeepRecursiveFunction] putting the call
activation frame on the heap.\n * @param [T] function parameter type.\n * @param [R] function result type.\n
*\n@RestrictsSuspension\n@SinceKotlin("1.7")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic
sealed class DeepRecursiveScope<T, R> {\n /**\n * Makes recursive call to this [DeepRecursiveFunction]
function putting the call activation frame on the heap,\n * as opposed to the actual call stack that is used by a
regular recursive call.\n *\n * public abstract suspend fun callRecursive(value: T): R\n *\n * Makes call
to the specified [DeepRecursiveFunction] function putting the call activation frame on the heap,\n * as opposed to
the actual call stack that is used by a regular call.\n *\n * public abstract suspend fun <U, S>
DeepRecursiveFunction<U, S>.callRecursive(value: U): S\n *\n @Deprecated(\n level =
DeprecationLevel.ERROR,\n message =\n "'invoke' should not be called from DeepRecursiveScope. '\n
+ \n
 '\n 'Use 'callRecursive' to do recursion in the heap instead of the call stack.'",\n replaceWith =
ReplaceWith("\n 'this.callRecursive(value)'\n)\n *)\n @Suppress("\n UNUSED_PARAMETER\n ")
public operator
fun DeepRecursiveFunction<*, *>.invoke(value: Any?): Nothing =\n throw
UnsupportedOperationException("Should not be called from DeepRecursiveScope")\n)\n)\n\n//
Implementation =====\n\nprivate typealias
DeepRecursiveFunctionBlock = suspend DeepRecursiveScope<*, *>.(Any?) ->
Any?\n\n@SharedImmutable\nprivate val UNDEFINED_RESULT =
Result.success(COROUTINE_SUSPENDED)\n\n@Suppress("\n UNCHECKED_CAST\n ")
private class
DeepRecursiveScopeImpl<T, R>(\n block: suspend DeepRecursiveScope<T, R>.(T) -> R,\n value: T)\n) :
DeepRecursiveScope<T, R>(), Continuation<R> {\n // Active function block\n private var function:
DeepRecursiveFunctionBlock = block as DeepRecursiveFunctionBlock\n\n // Value to call function with\n\n
private var value: Any? = value\n\n // Continuation of the current call\n private var cont: Continuation<Any?>?
= this as Continuation<Any?>\n\n // Completion result (completion of the whole call stack)\n private var result:
Result<Any?> = UNDEFINED_RESULT\n\n override val context: CoroutineContext\n get() =
EmptyCoroutineContext\n\n override fun resumeWith(result: Result<R>) {\n this.cont = null\n this.result
= result\n }\n\n override suspend fun callRecursive(value: T): R = suspendCoroutineUninterceptedOrReturn {
cont ->\n // calling the same function that is currently active\n this.cont = cont as Continuation<Any?>\n
this.value = value\n COROUTINE_SUSPENDED\n }\n\n override suspend fun <U, S>
DeepRecursiveFunction<U, S>.callRecursive(value: U): S = suspendCoroutineUninterceptedOrReturn { cont ->\n
 // calling another recursive function\n val function = block as DeepRecursiveFunctionBlock\n
with(this@DeepRecursiveScopeImpl) {\n val currentFunction = this.function\n if (function !==
currentFunction) {\n // calling a different function -- create a trampoline to restore function ref\n
this.function = function\n this.cont = crossFunctionCompletion(currentFunction, cont as
Continuation<Any?>)\n } else {\n // calling the same function -- direct\n this.cont = cont
as Continuation<Any?>\n }\n this.value = value\n }\n COROUTINE_SUSPENDED\n }\n\n
private fun crossFunctionCompletion(\n currentFunction: DeepRecursiveFunctionBlock,\n cont:
Continuation<Any?>\n) : Continuation<Any?> = Continuation(EmptyCoroutineContext) {\n this.function =

```

```

currentFunction\n // When going back from a trampoline we cannot just call cont.resume (stack usage!)\n //
We delegate the cont.resumeWith(it) call to runCallLoop\n this.cont = cont\n this.result = it\n }\n\n@Suppress("UNCHECKED_CAST")\n fun runCallLoop(): R {\n while (true) {\n // Note: cont is set
to null in DeepRecursiveScopeImpl.resumeWith when the whole computation completes\n val result =
this.result\n val cont = this.cont\n ?: return (result as Result<R>).getOrThrow() // done -- final
result\n // The order of comparison is important here for that case of rogue class with broken equals\n
if (UNDEFINED_RESULT == result) {\n // call "function" with "value" using "cont" as completion\n
 val r = try {\n // This is block.startCoroutine(this, value, cont)\n
function.startCoroutineUninterceptedOrReturn(this, value, cont)\n } catch (e: Throwable) {\n
cont.resumeWithException(e)\n continue\n }\n // If the function returns without
suspension -- calls its continuation immediately\n if (r !== COROUTINE_SUSPENDED)\n
cont.resume(r as R)\n } else {\n // we returned from a crossFunctionCompletion trampoline -- call
resume here\n this.result = UNDEFINED_RESULT // reset result back\n
cont.resumeWith(result)\n }\n }\n }\n\n"/\n\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n *^n\n// Auto-generated file. DO NOT
EDIT!\n\n@file:kotlin.jvm.JvmName("NumbersKt")\n@file:kotlin.jvm.JvmMultifileClass\npackage
kotlin\n\nimport kotlin.math.sign\n\n/** Divides this value by the other value, flooring the result to an integer that is
closer to negative infinity.
*\n\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Byte.floorDiv(other: Byte): Int = \n this.toInt().floorDiv(other.toInt())\n\n/** Calculates the remainder of
flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the
divisor and has the absolute value less than the absolute value of the divisor.\n
*\n\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Byte.mod(other: Byte): Byte = \n this.toInt().mod(other.toInt()).toByte()\n\n/** Divides this value by the other
value, flooring the result to an integer that is closer to negative infinity.
*\n\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Byte.floorDiv(other: Short): Int = \n this.toInt().floorDiv(other.toInt())\n\n/** Calculates the remainder of
flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the
divisor and has the absolute value less than the absolute value of the divisor.\n
*\n\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Byte.mod(other: Short): Short = \n this.toInt().mod(other.toInt()).toShort()\n\n/** Divides this value by the
other value, flooring the result to an integer that is closer to negative infinity.
*\n\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Byte.floorDiv(other: Int): Int = \n this.toInt().floorDiv(other)\n\n/** Calculates the remainder of flooring
division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and
has the absolute value less than the absolute value of the divisor.\n
*\n\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Byte.mod(other: Int): Int = \n this.toInt().mod(other)\n\n/** Divides this value by the other value, flooring the
result to an integer that is closer to negative infinity.
*\n\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Byte.floorDiv(other: Long): Long = \n this.toLong().floorDiv(other)\n\n/** Calculates the remainder of
flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the
divisor and has the absolute value less than the absolute value of the divisor.\n
*\n\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Byte.mod(other: Long): Long = \n this.toLong().mod(other)\n\n/** Divides this value by the other value,
flooring the result to an integer that is closer to negative infinity.
*\n\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline

```

```

fun Short.floorDiv(other: Byte): Int = \n this.toInt().floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of
flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the
divisor and has the absolute value less than the absolute value of the divisor.\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Short.mod(other: Byte): Byte = \n this.toInt().mod(other.toInt()).toByte()\n\n/** Divides this value by the
other value, flooring the result to an integer that is closer to negative infinity.
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Short.floorDiv(other: Short): Int = \n this.toInt().floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of
flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the
divisor and has the absolute value less than the absolute value of the divisor.\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Short.mod(other: Short): Short = \n this.toInt().mod(other.toInt()).toShort()\n\n/** Divides this value by the
other value, flooring the result to an integer that is closer to negative infinity.
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Short.floorDiv(other: Int): Int = \n this.toInt().floorDiv(other)\n\n/**\n * Calculates the remainder of flooring
division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and
has the absolute value less than the absolute value of the divisor.\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Short.mod(other: Int): Int = \n this.toInt().mod(other)\n\n/** Divides this value by the other value, flooring the
result to an integer that is closer to negative infinity.
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Short.floorDiv(other: Long): Long = \n this.toLong().floorDiv(other)\n\n/**\n * Calculates the remainder of
flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the
divisor and has the absolute value less than the absolute value of the divisor.\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Short.mod(other: Long): Long = \n this.toLong().mod(other)\n\n/** Divides this value by the other value,
flooring the result to an integer that is closer to negative infinity.
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Int.floorDiv(other: Byte): Int = \n this.floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of flooring
division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and
has the absolute value less than the absolute value of the divisor.\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Int.mod(other: Byte): Byte = \n this.mod(other.toInt()).toByte()\n\n/** Divides this value by the other value,
flooring the result to an integer that is closer to negative infinity.
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Int.floorDiv(other: Short): Int = \n this.floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of flooring
division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and
has the absolute value less than the absolute value of the divisor.\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Int.mod(other: Short): Short = \n this.mod(other.toInt()).toShort()\n\n/** Divides this value by the other value,
flooring the result to an integer that is closer to negative infinity.
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Int.floorDiv(other: Int): Int {\n var q = this / other\n if (this xor other < 0 && q * other != this) q--\n return
q}\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is
either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the
divisor.\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline
fun Int.mod(other: Int): Int {\n val r = this % other\n return r + (other and (((r xor other) and (r or -r)) shr

```

31))\n\n/\*\* Divides this value by the other value, flooring the result to an integer that is closer to negative infinity. \*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Int.floorDiv(other: Long): Long = \n this.toLong().floorDiv(other)\n\n/\*\*\n \* Calculates the remainder of flooring division of this value by the other value.\n \* \n \* The result is either zero or has the same sign as the \_divisor\_ and has the absolute value less than the absolute value of the divisor.\n \*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Int.mod(other: Long): Long = \n this.toLong().mod(other)\n\n/\*\* Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n \*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.floorDiv(other: Byte): Long = \n this.floorDiv(other.toLong())\n\n/\*\*\n \* Calculates the remainder of flooring division of this value by the other value.\n \* \n \* The result is either zero or has the same sign as the \_divisor\_ and has the absolute value less than the absolute value of the divisor.\n \*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.mod(other: Byte): Byte = \n this.mod(other.toLong()).toByte()\n\n/\*\* Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n \*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.floorDiv(other: Short): Long = \n this.floorDiv(other.toLong())\n\n/\*\*\n \* Calculates the remainder of flooring division of this value by the other value.\n \* \n \* The result is either zero or has the same sign as the \_divisor\_ and has the absolute value less than the absolute value of the divisor.\n \*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.mod(other: Short): Short = \n this.mod(other.toLong()).toShort()\n\n/\*\* Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n \*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.floorDiv(other: Int): Long = \n this.floorDiv(other.toLong())\n\n/\*\*\n \* Calculates the remainder of flooring division of this value by the other value.\n \* \n \* The result is either zero or has the same sign as the \_divisor\_ and has the absolute value less than the absolute value of the divisor.\n \*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.mod(other: Int): Int = \n this.mod(other.toLong()).toInt()\n\n/\*\* Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n \*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.floorDiv(other: Long): Long {\n var q = this / other\n if (this xor other < 0 && q \* other != this) q--\n return q}\n\n/\*\*\n \* Calculates the remainder of flooring division of this value by the other value.\n \* \n \* The result is either zero or has the same sign as the \_divisor\_ and has the absolute value less than the absolute value of the divisor.\n \*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.mod(other: Long): Long {\n val r = this % other\n return r + (other and (((r xor other) and (r or -r)) shr 63))}\n\n/\*\*\n \* Calculates the remainder of flooring division of this value by the other value.\n \* \n \* The result is either zero or has the same sign as the \_divisor\_ and has the absolute value less than the absolute value of the divisor.\n \* \n \* If the result cannot be represented exactly, it is rounded to the nearest representable number. In this case the absolute value of the result can be less than or \_equal to\_ the absolute value of the divisor.\n \*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Float.mod(other: Float): Float {\n val r = this % other\n return if (r != 0.0.toFloat() && r.sign != other.sign) r + other else r}\n\n/\*\*\n \* Calculates the remainder of flooring division of this value by the other value.\n \* \n \* The result is either zero or has the same sign as the \_divisor\_ and has the absolute value less than the absolute value of the divisor.\n \* \n \* If the result cannot be represented exactly, it is rounded to the nearest representable number. In this case the absolute value of the result can be less than or \_equal to\_ the absolute value of the divisor.\n \*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Float.mod(other: Double): Double = \n this.toDouble().mod(other)\n\n/\*\*\n \* Calculates the remainder of

flooring division of this value by the other value.  
 \* The result is either zero or has the same sign as the `_divisor_` and has the absolute value less than the absolute value of the divisor.  
 \* If the result cannot be represented exactly, it is rounded to the nearest representable number. In this case the absolute value of the result can be less than or `_equal to_` the absolute value of the divisor.

```

*\/n@SinceKotlin("1.5")n@kotlin.internal.InlineOnlyn@kotlin.internal.IntrinsicConstEvaluationnpublic inline
fun Double.mod(other: Float): Double = \n this.mod(other.toDouble())n/**n * Calculates the remainder of
flooring division of this value by the other value.n * The result is either zero or has the same sign as the
divisor and has the absolute value less than the absolute value of the divisor.n * If the result cannot be
represented exactly, it is rounded to the nearest representable number. In this case the absolute value of the result
can be less than or _equal to_ the absolute value of the divisor.\n
*\/n@SinceKotlin("1.5")n@kotlin.internal.InlineOnlyn@kotlin.internal.IntrinsicConstEvaluationnpublic inline
fun Double.mod(other: Double): Double {n val r = this % other\n return if (r != 0.0 && r.sign != other.sign) r +
other else r}n"/**n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n
* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\/n\npackage kotlin\n\nimport kotlin.internal.InlineOnly\n\n/**n * Returns a hash code value for the object or
zero if the object is `null`.n * @see Any.hashCode\n *\/n@SinceKotlin("1.3")n@InlineOnlynpublic inline
fun Any?.hashCode(): Int = this?.hashCode() ?: 0n"/**n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n *\/n\npackage kotlin\n\n/**n * Represents a version of the Kotlin standard
library.n * [major], [minor] and [patch] are integer components of a version,\n * they must be non-negative and
not greater than 255 (MAX_COMPONENT_VALUE).\n * @constructor Creates a version from all three
components.n *\/n@SinceKotlin("1.1")npublic class KotlinVersion(val major: Int, val minor: Int, val patch: Int) :
Comparable<KotlinVersion> {n /**n * Creates a version from [major] and [minor] components, leaving
[patch] component zero.n *\/n public constructor(major: Int, minor: Int) : this(major, minor, 0)\n\n private val
version = versionOf(major, minor, patch)\n\n private fun versionOf(major: Int, minor: Int, patch: Int): Int {n
require(major in 0..MAX_COMPONENT_VALUE && minor in 0..MAX_COMPONENT_VALUE && patch in
0..MAX_COMPONENT_VALUE) {n \nVersion components are out of range: $major.$minor.$patch\n }\n return major.shl(16) + minor.shl(8) + patch\n }\n\n /**n * Returns the string representation of this
version\n *\/n override fun toString(): String = "$major.$minor.$patch"\n\n override fun equals(other:
Any?): Boolean {n if (this === other) return true\n val otherVersion = (other as? KotlinVersion) ?: return
false\n return this.version == otherVersion.version\n }\n\n override fun hashCode(): Int = version\n\n override fun compareTo(other: KotlinVersion): Int = version - other.version\n\n /**n * Returns `true` if this
version is not less than the version specified\n * with the provided [major] and [minor] components.n *\/n
public fun isAtLeast(major: Int, minor: Int): Boolean = // this.version >= versionOf(major, minor, 0)\n
this.major > major || (this.major == major &&\n this.minor >= minor)\n\n /**n * Returns `true` if this
version is not less than the version specified\n * with the provided [major], [minor] and [patch] components.\n *\/n
public fun isAtLeast(major: Int, minor: Int, patch: Int): Boolean = // this.version >= versionOf(major, minor,
patch)\n this.major > major || (this.major == major &&\n (this.minor > minor || this.minor == minor
&&\n this.patch >= patch))\n\n companion object {n /**n * Maximum value a version
component can have, a constant value 255.\n *\/n // NOTE: Must be placed before CURRENT because its
initialization requires this field being initialized in JS\n public const val MAX_COMPONENT_VALUE =
255\n\n /**n * Returns the current version of the Kotlin standard library.\n *\/n
@kotlin.jvm.JvmField\n public val CURRENT: KotlinVersion = KotlinVersionCurrentValue.get()\n }\n}\n\n// this class is ignored during classpath normalization when considering whether to recompile dependencies
in Kotlin build\nprivate object KotlinVersionCurrentValue {n @kotlin.jvm.JvmStatic\n fun get():
KotlinVersion = KotlinVersion(1, 7, 20) // value is written here automatically during build\n"/**n * Copyright
2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n

```

```

*\n\n@file:kotlin.jvm.JvmName("LateinitKt")\n@file:Suppress("unused")\n\npackage kotlin\n\nimport
kotlin.internal.InlineOnly\nimport kotlin.internal.AccessibleLateinitPropertyLiteral\nimport
kotlin.reflect.KProperty0\n\n/**\n * Returns `true` if this lateinit property has been assigned a value, and `false`
otherwise.\n *\n * Cannot be used in an inline function, to avoid binary compatibility issues.\n
*\n\n@SinceKotlin("1.2")\n@InlineOnly\n\ninline val @receiver:AccessibleLateinitPropertyLiteral
KProperty0<*>.isInitialized: Boolean\n get() = throw NotImplementedError("Implementation is
intrinsic")\n","/**\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use
of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmName("LazyKt")\n@file:kotlin.jvm.JvmMultifileClass\n\npackage kotlin\n\nimport
kotlin.reflect.KProperty\n\n/**\n * Represents a value with lazy initialization.\n *\n * To create an instance of
[Lazy] use the [lazy] function.\n *\n\npublic interface Lazy<out T> {\n /**\n * Gets the lazily initialized value of
the current Lazy instance.\n *\n * Once the value was initialized it must not change during the rest of lifetime of this
Lazy instance.\n *\n * public val value: T\n */\n /**\n * Returns `true` if a value for this Lazy instance has been
already initialized, and `false` otherwise.\n *\n * Once this function has returned `true` it stays `true` for the rest of
lifetime of this Lazy instance.\n *\n * public fun isInitialized(): Boolean\n}\n\n/**\n * Creates a new instance of
the [Lazy] that is already initialized with the specified [value].\n *\n\npublic fun <T> lazyOf(value: T): Lazy<T> =
InitializedLazyImpl(value)\n\n/**\n * An extension to delegate a read-only property of type [T] to an instance of
[Lazy].\n *\n * This extension allows to use instances of Lazy for property delegation:\n *\n`val property: String by
lazy { initializer }`\n *\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun <T> Lazy<T>.getValue(thisRef:
Any?, property: KProperty<*>): T = value\n\n/**\n * Specifies how a [Lazy] instance synchronizes initialization
among multiple threads.\n *\n\npublic enum class LazyThreadSafetyMode {\n\n /**\n * Locks are used to ensure
that only a single thread can initialize the [Lazy] instance.\n *\n * SYNCHRONIZED,\n */\n /**\n * Initializer
function can be called several times on concurrent access to uninitialized [Lazy] instance value,\n * but only the
first returned value will be used as the value of [Lazy] instance.\n *\n * PUBLICATION,\n */\n /**\n * No
locks are used to synchronize an access to the [Lazy] instance value; if the instance is accessed from multiple
threads, its behavior is undefined.\n *\n * This mode should not be used unless the [Lazy] instance is
guaranteed never to be initialized from more than one thread.\n *\n * NONE,\n */\n}\n\n\ninternal object
UNINITIALIZED_VALUE\n\n// internal to be called from lazy in JS\n\ninternal class UnsafeLazyImpl<out
T>(initializer: () -> T) : Lazy<T>, Serializable {\n private var initializer: (() -> T)? = initializer\n private var
_value: Any? = UNINITIALIZED_VALUE\n\n override val value: T\n get() {\n if (_value ===
UNINITIALIZED_VALUE) {\n _value = initializer!!()\n initializer = null\n }\n
}\n\n @Suppress("UNCHECKED_CAST")\n return _value as T\n }\n\n override fun isInitialized():
Boolean = _value !== UNINITIALIZED_VALUE\n\n override fun toString(): String = if (isInitialized())
value.toString() else "Lazy value not initialized yet."\n\n private fun writeReplace(): Any =
InitializedLazyImpl(value)\n}\n\n\ninternal class InitializedLazyImpl<out T>(override val value: T) : Lazy<T>,
Serializable {\n\n override fun isInitialized(): Boolean = true\n\n override fun toString(): String =
value.toString()\n}\n","/**\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("NumbersKt")\n\npackage kotlin\n\n/**\n *
Counts the number of set bits in the binary representation of this [Int] number.\n
*\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n\npublic expect fun
Int.countOneBits(): Int\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the binary
representation of this [Int] number.\n
*\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n\npublic expect fun
Int.countLeadingZeroBits(): Int\n\n/**\n * Counts the number of consecutive least significant bits that are zero in
the binary representation of this [Int] number.\n
*\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n\npublic expect fun

```



Int.countTrailingZeroBits(): Int\n\n/\*\*\n \* Returns a number having a single bit set in the position of the most significant set bit of this [Int] number,\n \* or zero, if this number is zero.\n

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
```

Int.takeHighestOneBit(): Int\n\n/\*\*\n \* Returns a number having a single bit set in the position of the least significant set bit of this [Int] number,\n \* or zero, if this number is zero.\n

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
```

Int.takeLowestOneBit(): Int\n\n/\*\*\n \* Rotates the binary representation of this [Int] number left by the specified [bitCount] number of bits.\n \* The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n \* Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n \* `number.rotateLeft(-n) == number.rotateRight(n)`\n \* Rotating by a multiple of [Int.SIZE\_BITS] (32) returns the same number, or more generally\n \* `number.rotateLeft(n) == number.rotateLeft(n % 32)`\n

```
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
```

Int.rotateLeft(bitCount: Int): Int\n\n/\*\*\n \* Rotates the binary representation of this [Int] number right by the specified [bitCount] number of bits.\n \* The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n \* Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n \* `number.rotateRight(-n) == number.rotateLeft(n)`\n \* Rotating by a multiple of [Int.SIZE\_BITS] (32) returns the same number, or more generally\n \* `number.rotateRight(n) == number.rotateRight(n % 32)`\n

```
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
```

Int.rotateRight(bitCount: Int): Int\n\n/\*\*\n \* Counts the number of set bits in the binary representation of this [Long] number.\n

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
```

Long.countOneBits(): Int\n\n/\*\*\n \* Counts the number of consecutive most significant bits that are zero in the binary representation of this [Long] number.\n

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
```

Long.countLeadingZeroBits(): Int\n\n/\*\*\n \* Counts the number of consecutive least significant bits that are zero in the binary representation of this [Long] number.\n

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
```

Long.countTrailingZeroBits(): Int\n\n/\*\*\n \* Returns a number having a single bit set in the position of the most significant set bit of this [Long] number,\n \* or zero, if this number is zero.\n

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
```

Long.takeHighestOneBit(): Long\n\n/\*\*\n \* Returns a number having a single bit set in the position of the least significant set bit of this [Long] number,\n \* or zero, if this number is zero.\n

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
```

Long.takeLowestOneBit(): Long\n\n/\*\*\n \* Rotates the binary representation of this [Long] number left by the specified [bitCount] number of bits.\n \* The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n \* Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n \* `number.rotateLeft(-n) == number.rotateRight(n)`\n \* Rotating by a multiple of [Long.SIZE\_BITS] (64) returns the same number, or more generally\n \* `number.rotateLeft(n) == number.rotateLeft(n % 64)`\n

```
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
```

Long.rotateLeft(bitCount: Int): Long\n\n/\*\*\n \* Rotates the binary representation of this [Long] number right by the specified [bitCount] number of bits.\n \* The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n \* Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n \* `number.rotateRight(-n) == number.rotateLeft(n)`\n \* Rotating by a multiple of [Long.SIZE\_BITS] (64) returns the same number, or more generally\n \* `number.rotateRight(n) == number.rotateRight(n % 64)`\n

```
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
```

Long.rotateRight(bitCount: Int): Long\n\n\*\*\n \* Counts the number of set bits in the binary representation of this [Byte] number.\n

\*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.countOneBits(): Int = (toInt() and 0xFF).countOneBits()\n\n\*\*\n \* Counts the number of consecutive most significant bits that are zero in the binary representation of this [Byte] number.\n

\*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.countLeadingZeroBits(): Int = (toInt() and 0xFF).countLeadingZeroBits() - (Int.SIZE\_BITS - Byte.SIZE\_BITS)\n\n\*\*\n \* Counts the number of consecutive least significant bits that are zero in the binary representation of this [Byte] number.\n

\*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.countTrailingZeroBits(): Int = (toInt() or 0x100).countTrailingZeroBits()\n\n\*\*\n \* Returns a number having a single bit set in the position of the most significant set bit of this [Byte] number,\n \* or zero, if this number is zero.\n

\*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.takeHighestOneBit(): Byte = (toInt() and 0xFF).takeHighestOneBit().toByte()\n\n\*\*\n \* Returns a number having a single bit set in the position of the least significant set bit of this [Byte] number,\n \* or zero, if this number is zero.\n

\*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.takeLowestOneBit(): Byte = toInt().takeLowestOneBit().toByte()\n\n\*\*\n \* Rotates the binary representation of this [Byte] number left by the specified [bitCount] number of bits.\n \* The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n \* Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n \* `number.rotateLeft(-n) == number.rotateRight(n)`\n \* Rotating by a multiple of [Byte.SIZE\_BITS] (8) returns the same number, or more generally\n \* `number.rotateLeft(n) == number.rotateLeft(n % 8)`\n

\*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Byte.rotateLeft(bitCount: Int): Byte =\n (toInt().shl(bitCount and 7) or (toInt() and 0xFF).ushr(8 - (bitCount and 7))).toByte()\n\n\*\*\n \* Rotates the binary representation of this [Byte] number right by the specified [bitCount] number of bits.\n \* The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n \* Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n \* `number.rotateRight(-n) == number.rotateLeft(n)`\n \* Rotating by a multiple of [Byte.SIZE\_BITS] (8) returns the same number, or more generally\n \* `number.rotateRight(n) == number.rotateRight(n % 8)`\n

\*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Byte.rotateRight(bitCount: Int): Byte =\n (toInt().shl(8 - (bitCount and 7)) or (toInt() and 0xFF).ushr(bitCount and 7)).toByte()\n\n\*\*\n \* Counts the number of set bits in the binary representation of this [Short] number.\n

\*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.countOneBits(): Int = (toInt() and 0xFFFF).countOneBits()\n\n\*\*\n \* Counts the number of consecutive most significant bits that are zero in the binary representation of this [Short] number.\n

\*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.countLeadingZeroBits(): Int =\n (toInt() and 0xFFFF).countLeadingZeroBits() - (Int.SIZE\_BITS - Short.SIZE\_BITS)\n\n\*\*\n \* Counts the number of consecutive least significant bits that are zero in the binary representation of this [Short] number.\n

\*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.countTrailingZeroBits(): Int = (toInt() or 0x10000).countTrailingZeroBits()\n\n\*\*\n \* Returns a number having a single bit set in the position of the most significant set bit of this [Short] number,\n \* or zero, if this number is zero.\n

\*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.takeHighestOneBit(): Short = (toInt() and 0xFFFF).takeHighestOneBit().toShort()\n\n\*\*\n \*

Returns a number having a single bit set in the position of the least significant set bit of this [Short] number, or zero, if this number is zero.

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.takeLowestOneBit(): Short = toInt().takeLowestOneBit().toShort()\n\n/**\n * Rotates the binary representation of this [Short] number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n * Rotating by a multiple of [Short.SIZE_BITS] (16) returns the same number, or more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 16)`\n
```

```
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Short.rotateLeft(bitCount: Int): Short =\n (toInt().shl(bitCount and 15) or (toInt() and 0xFFFF).ushr(16 - (bitCount and 15))).toShort()\n\n/**\n * Rotates the binary representation of this [Short] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n * Rotating by a multiple of [Short.SIZE_BITS] (16) returns the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 16)`\n
```

```
\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Short.rotateRight(bitCount: Int): Short =\n (toInt().shl(16 - (bitCount and 15)) or (toInt() and 0xFFFF).ushr(bitCount and 15)).toShort()\n\n"/\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\npackage kotlin\nimport kotlin.internal.RequireKotlin\nimport
```

```
kotlin.internal.RequireKotlinVersionKind\n@kotlin.internal.InlineOnly\n@SinceKotlin("1.2")\n@Suppress("INVISIBLE_MEMBER", "INVISIBLE_REFERENCE")\npublic inline fun <R> suspend(noinline block: suspend () -> R): suspend () -> R = block\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n@file:kotlin.jvm.JvmName("TuplesKt")\npackage kotlin\n\n/**\n
```

Represents a generic pair of two values. There is no meaning attached to values in this class, it can be used for any purpose. Pair exhibits value semantics, i.e. two pairs are equal if both components are equal. An example of decomposing it into values: @sample samples.misc.Tuples.pairDestructuring\n\n @param A type of the first value.\n @param B type of the second value.\n @property first First value.\n @property second Second value.\n @constructor Creates a new instance of Pair.\n\npublic data class Pair<out A, out B>(\n

```
 public val first: A,\n public val second: B\n) : Serializable {\n\n /**\n * Returns string representation of the [Pair] including its [first] and [second] values.\n *\n * public override fun toString(): String = \"($first, $second)\"\n }\n\n /**\n * Creates a tuple of type [Pair] from this and [that].\n *\n * This can be useful for creating [Map] literals with less noise, for example:\n *\n * @sample samples.collections.Maps.Instantiation.mapFromPairs\n */\n\n public infix fun <A, B> A.to(that: B): Pair<A, B> = Pair(this, that)\n\n /**\n * Converts this pair into a list.\n *\n * @sample samples.misc.Tuples.pairToList\n */\n\n public fun <T> Pair<T, T>.toList(): List<T> = listOf(first,
```

```
second)\n\n/**\n * Represents a triad of values\n * There is no meaning attached to values in this class, it can be used for any purpose.\n * Triple exhibits value semantics, i.e. two triples are equal if all three components are equal.\n * An example of decomposing it into values:\n * @sample samples.misc.Tuples.tripleDestructuring\n *\n * @param A type of the first value.\n * @param B type of the second value.\n * @param C type of the third value.\n *\n * @property first First value.\n * @property second Second value.\n * @property third Third value.\n */\n\npublic data class Triple<out A, out B, out C>(\n
```

```
 public val first: A,\n public val second: B,\n public val third: C\n) : Serializable {\n\n /**\n * Returns string representation of the [Triple] including its [first], [second] and [third] values.\n *\n * public override fun toString(): String = \"($first, $second, $third)\"\n }\n\n /**\n * Converts this triple into a list.\n *\n * @sample samples.misc.Tuples.tripleToList\n */\n\n public fun <T> Triple<T, T, T>.toList(): List<T> = listOf(first, second, third)\n\n"/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming
```

```

Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n *\/\n\n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin.ranges\n\nimport
kotlin.internal.*\n\n/**\n * A range of values of type `UInt`.\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@OptIn(ExperimentalStdlib
Api::class)\npublic class UIntRange(start: UInt, endInclusive: UInt) : UIntProgression(start, endInclusive, 1),
ClosedRange<UInt>, OpenEndRange<UInt> {\n override val start: UInt get() = first\n override val endInclusive:
UInt get() = last\n \n @SinceKotlin("1.7")\n @ExperimentalStdlibApi\n @Deprecated("Can throw an
exception when it's impossible to represent the value with UInt type, for example, when the range includes
MAX_VALUE. It's recommended to use 'endInclusive' property that doesn't throw.")\n override val
endExclusive: UInt get() {\n if (last == UInt.MAX_VALUE) error("Cannot return the exclusive upper bound
of a range that includes MAX_VALUE.")\n return last + 1\n }\n override fun contains(value: UInt):
Boolean = first <= value && value <= last\n /**\n * Checks if the range is empty.\n * \n * The range is
empty if its start value is greater than the end value.\n */\n override fun isEmpty(): Boolean = first > last\n override fun equals(other: Any?): Boolean =\n other is UIntRange && (isEmpty() && other.isEmpty()) ||\n first == other.first && last == other.last\n override fun hashCode(): Int =\n if (isEmpty()) -1 else (31 *
first.toInt() + last.toInt())\n override fun toString(): String = "$first..$last"\n companion object {\n /**\n * An empty range of values of type UInt.\n */\n public val EMPTY: UIntRange = UIntRange(UInt.MAX_VALUE,
UInt.MIN_VALUE)\n }\n}\n\n/**\n * A progression of values of type `UInt`.\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic open class
UIntProgression\n\ninternal constructor(\n start: UInt,\n endInclusive: UInt,\n step: Int\n) : Iterable<UInt> {\n
 init {\n if (step == 0.toInt()) throw kotlin.IllegalArgumentException("Step must be non-zero.")\n if (step
== Int.MIN_VALUE) throw kotlin.IllegalArgumentException("Step must be greater than Int.MIN_VALUE to
avoid overflow on negation.")\n }\n /**\n * The first element in the progression.\n */\n public val first:
UInt = start\n /**\n * The last element in the progression.\n */\n public val last: UInt =
getProgressionLastElement(start, endInclusive, step)\n /**\n * The step of the progression.\n */\n public
val step: Int = step\n final override fun iterator(): Iterator<UInt> = UIntProgressionIterator(first, last, step)\n /**\n * Checks if the progression is empty.\n * \n * Progression with a positive step is empty if its first
element is greater than the last element.\n * \n * Progression with a negative step is empty if its first element is less
than the last element.\n */\n public open fun isEmpty(): Boolean = if (step > 0) first > last else first < last\n override fun equals(other: Any?): Boolean =\n other is UIntProgression && (isEmpty() && other.isEmpty()) ||\n first == other.first && last == other.last && step == other.step)\n override fun hashCode(): Int =\n if (isEmpty()) -1 else (31 * (31 * first.toInt() + last.toInt()) + step.toInt())\n override fun toString(): String = if
(step > 0) "$first..$last step $step" else "$first downTo $last step ${-step}"\n companion object {\n /**\n * Creates UIntProgression within the specified bounds of a closed range.\n * \n * The progression starts with
the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].\n * \n * In
order to go backwards the [step] must be negative.\n * \n * [step] must be greater than `Int.MIN_VALUE`
and not equal to zero.\n */\n public fun fromClosedRange(rangeStart: UInt, rangeEnd: UInt, step: Int):
UIntProgression = UIntProgression(rangeStart, rangeEnd, step)\n }\n}\n\n/**\n * An iterator over a progression
of values of type `UInt`.\n * \n * @property step the number by which the value is incremented on each step.\n
*\n\n@SinceKotlin("1.3")\nprivate class UIntProgressionIterator(first: UInt, last: UInt, step: Int) : Iterator<UInt>
{\n private val finalElement = last\n private var hasNext: Boolean = if (step > 0) first <= last else first >= last\n
 private val step = step.toInt() // use 2-complement math for negative steps\n private var next = if (hasNext) first
else finalElement\n override fun hasNext(): Boolean = hasNext\n override fun next(): UInt {\n val value
= next\n if (value == finalElement) {\n if (!hasNext) throw kotlin.NoSuchElementException()\n hasNext = false\n } else {\n next += step\n }\n return value\n }\n}\n\n"/**\n * Copyright
2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\/\n\n// Auto-generated file. DO NOT
EDIT!\n\npackage kotlin.ranges\n\nimport kotlin.internal.*\n\n/**\n * A range of values of type `ULong`.\n

```

```

*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@OptIn(ExperimentalStdlib
Api::class)\npublic class ULongRange(start: ULong, endInclusive: ULong) : ULongProgression(start, endInclusive,
1), ClosedRange<ULong>, OpenEndRange<ULong> {\n override val start: ULong get() = first\n override val
endInclusive: ULong get() = last\n \n @SinceKotlin("1.7")\n @ExperimentalStdlibApi\n @Deprecated("Can throw an exception when it's impossible to represent the value with ULong type, for example,
when the range includes MAX_VALUE. It's recommended to use 'endInclusive' property that doesn't throw.")\n override val endExclusive: ULong get() {\n if (last == ULong.MAX_VALUE) error("Cannot return the
exclusive upper bound of a range that includes MAX_VALUE.")\n return last + 1u\n }\n \n override fun
contains(value: ULong): Boolean = first <= value && value <= last\n \n /** \n * Checks if the range is empty.\n \n * The range is empty if its start value is greater than the end value.\n */\n override fun isEmpty(): Boolean
= first > last\n \n override fun equals(other: Any?): Boolean =\n other is ULongRange && (isEmpty() &&
other.isEmpty()) ||\n first == other.first && last == other.last\n \n override fun hashCode(): Int =\n if
(isEmpty()) -1 else (31 * (first xor (first shr 32)).toInt() + (last xor (last shr 32)).toInt())\n \n override fun
toString(): String = "$first..$last"\n \n companion object {\n /** An empty range of values of type ULong.
*/\n public val EMPTY: ULongRange = ULongRange(ULong.MAX_VALUE, ULong.MIN_VALUE)\n }\n \n /**\n * A progression of values of type `ULong`.
/\n}\n\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic open class
ULongProgression\n internal constructor(\n start: ULong,\n endInclusive: ULong,\n step: Long\n) :
Iterable<ULong> {\n init {\n if (step == 0.toLong()) throw kotlin.IllegalArgumentException("Step must be
non-zero.")\n if (step == Long.MIN_VALUE) throw kotlin.IllegalArgumentException("Step must be greater
than Long.MIN_VALUE to avoid overflow on negation.")\n }\n \n /**\n * The first element in the
progression.\n */\n public val first: ULong = start\n \n /**\n * The last element in the progression.\n */\n public val last: ULong = getProgressionLastElement(start, endInclusive, step)\n \n /**\n * The step of the
progression.\n */\n public val step: Long = step\n \n final override fun iterator(): Iterator<ULong> =
ULongProgressionIterator(first, last, step)\n \n /** \n * Checks if the progression is empty.\n \n * Progression with a positive step is empty if its first element is greater than the last element.\n * Progression with a
negative step is empty if its first element is less than the last element.\n */\n public open fun isEmpty(): Boolean
= if (step > 0) first > last else first < last\n \n override fun equals(other: Any?): Boolean =\n other is
ULongProgression && (isEmpty() && other.isEmpty()) ||\n first == other.first && last == other.last &&
step == other.step)\n \n override fun hashCode(): Int =\n if (isEmpty()) -1 else (31 * (31 * (first xor (first shr
32)).toInt() + (last xor (last shr 32)).toInt()) + (step xor (step ushr 32)).toInt())\n \n override fun toString(): String =
if (step > 0) "$first..$last step $step" else "$first downTo $last step ${-step}"\n \n companion object {\n /**\n * Creates ULongProgression within the specified bounds of a closed range.\n \n * The progression
starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].\n \n * In order to go backwards the [step] must be negative.\n \n * [step] must be greater than
`Long.MIN_VALUE` and not equal to zero.\n */\n public fun fromClosedRange(rangeStart: ULong,
rangeEnd: ULong, step: Long): ULongProgression = ULongProgression(rangeStart, rangeEnd, step)\n }\n \n /**\n * An iterator over a progression of values of type `ULong`. \n * @property step the number by which
the value is incremented on each step.\n */\n}\n\n*\n@SinceKotlin("1.3")\nprivate class ULongProgressionIterator(first:
ULong, last: ULong, step: Long) : Iterator<ULong> {\n private val finalElement = last\n private var hasNext:
Boolean = if (step > 0) first <= last else first >= last\n private val step = step.toULong() // use 2-complement math
for negative steps\n private var next = if (hasNext) first else finalElement\n \n override fun hasNext(): Boolean =
hasNext\n \n override fun next(): ULong {\n val value = next\n if (value == finalElement) {\n if
(!hasNext) throw kotlin.NoSuchElementException()\n hasNext = false\n } else {\n next += step\n }\n return value\n }\n}\n\n"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.math\n\n/**\n * Returns the smaller of two values.\n */\n}\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\

```

```

public inline fun min(a: UInt, b: UInt): UInt { \n return minOf(a, b)\n}\n\n/**\n * Returns the smaller of two values.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun min(a: ULong, b: ULong): ULong {\n return minOf(a, b)\n}\n\n/**\n * Returns the greater of two values.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun max(a: UInt, b: UInt): UInt {\n return maxOf(a, b)\n}\n\n/**\n * Returns the greater of two values.\n
\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun max(a: ULong, b: ULong): ULong {\n return maxOf(a, b)\n}", "/\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmName("UNumbersKt")\npackage kotlin\n\n/**\n * Counts the number of set bits in the binary representation of this [UInt] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.countOneBits(): Int = toInt().countOneBits()\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the binary representation of this [UInt] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.countLeadingZeroBits(): Int = toInt().countLeadingZeroBits()\n\n/**\n * Counts the number of consecutive least significant bits that are zero in the binary representation of this [UInt] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.countTrailingZeroBits(): Int = toInt().countTrailingZeroBits()\n\n/**\n * Returns a number having a single bit set in the position of the most significant set bit of this [UInt] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.takeHighestOneBit(): UInt = toInt().takeHighestOneBit().toUInt()\n\n/**\n * Returns a number having a single bit set in the position of the least significant set bit of this [UInt] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.takeLowestOneBit(): UInt = toInt().takeLowestOneBit().toUInt()\n\n/**\n * Rotates the binary representation of this [UInt] number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n * Rotating by a multiple of [UInt.SIZE_BITS] (32) returns the same number, or more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 32)`\n *\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class, ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.rotateLeft(bitCount: Int): UInt = toInt().rotateLeft(bitCount).toUInt()\n\n/**\n * Rotates the binary representation of this [UInt] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n * Rotating by a multiple of [UInt.SIZE_BITS] (32) returns the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 32)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class, ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.rotateRight(bitCount: Int): UInt = toInt().rotateRight(bitCount).toUInt()\n\n/**\n * Counts the number of set bits in the binary representation

```

of this [ULong] number.\n \*  
 @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n @kotlin.internal.InlineOnly\n public inline fun ULong.countOneBits(): Int = toLong().countOneBits()\n /\*\*\n \* Counts the number of consecutive most significant bits that are zero in the binary representation of this [ULong] number.\n \*/\n \*  
 @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n @kotlin.internal.InlineOnly\n public inline fun ULong.countLeadingZeroBits(): Int = toLong().countLeadingZeroBits()\n /\*\*\n \* Counts the number of consecutive least significant bits that are zero in the binary representation of this [ULong] number.\n \*/\n \*  
 @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n @kotlin.internal.InlineOnly\n public inline fun ULong.countTrailingZeroBits(): Int = toLong().countTrailingZeroBits()\n /\*\*\n \* Returns a number having a single bit set in the position of the most significant set bit of this [ULong] number,\n \* or zero, if this number is zero.\n \*/\n \*  
 @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n @kotlin.internal.InlineOnly\n public inline fun ULong.takeHighestOneBit(): ULong = toLong().takeHighestOneBit().toULong()\n /\*\*\n \* Returns a number having a single bit set in the position of the least significant set bit of this [ULong] number,\n \* or zero, if this number is zero.\n \*/\n \*  
 @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n @kotlin.internal.InlineOnly\n public inline fun ULong.takeLowestOneBit(): ULong = toLong().takeLowestOneBit().toULong()\n /\*\*\n \* Rotates the binary representation of this [ULong] number left by the specified [bitCount] number of bits.\n \* The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n \* Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n \* `number.rotateLeft(-n) == number.rotateRight(n)`\n \* Rotating by a multiple of [ULong.SIZE\_BITS] (64) returns the same number, or more generally\n \* `number.rotateLeft(n) == number.rotateLeft(n % 64)`\n \*/\n \*  
 @SinceKotlin("1.6")\n @WasExperimental(ExperimentalStdlibApi::class, ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n public inline fun ULong.rotateLeft(bitCount: Int): ULong = toLong().rotateLeft(bitCount).toULong()\n /\*\*\n \* Rotates the binary representation of this [ULong] number right by the specified [bitCount] number of bits.\n \* The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n \* Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n \* `number.rotateRight(-n) == number.rotateLeft(n)`\n \* Rotating by a multiple of [ULong.SIZE\_BITS] (64) returns the same number, or more generally\n \* `number.rotateRight(n) == number.rotateRight(n % 64)`\n \*/\n \*  
 @SinceKotlin("1.6")\n @WasExperimental(ExperimentalStdlibApi::class, ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n public inline fun ULong.rotateRight(bitCount: Int): ULong = toLong().rotateRight(bitCount).toULong()\n /\*\*\n \* Counts the number of set bits in the binary representation of this [UByte] number.\n \*/\n \*  
 @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n @kotlin.internal.InlineOnly\n public inline fun UByte.countOneBits(): Int = toUInt().countOneBits()\n /\*\*\n \* Counts the number of consecutive most significant bits that are zero in the binary representation of this [UByte] number.\n \*/\n \*  
 @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n @kotlin.internal.InlineOnly\n public inline fun UByte.countLeadingZeroBits(): Int = toByte().countLeadingZeroBits()\n /\*\*\n \* Counts the number of consecutive least significant bits that are zero in the binary representation of this [UByte] number.\n \*/\n \*  
 @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n @kotlin.internal.InlineOnly\n public inline fun UByte.countTrailingZeroBits(): Int = toByte().countTrailingZeroBits()\n /\*\*\n \* Returns a number having a single bit set in the position of the most significant set bit of this [UByte] number,\n \* or zero, if this number is zero.\n \*/

```

*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.takeHighestOneBit(): UByte
= toInt().takeHighestOneBit().toUByte()\n\n/**\n * Returns a number having a single bit set in the position of the
least significant set bit of this [UByte] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.takeLowestOneBit(): UByte =
toInt().takeLowestOneBit().toUByte()\n\n/**\n * Rotates the binary representation of this [UByte] number left by
the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the
number as the least significant bits on the right side.\n * Rotating the number left by a negative bit count is the
same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n *
Rotating by a multiple of [UByte.SIZE_BITS] (8) returns the same number, or more generally\n *
`number.rotateLeft(n) == number.rotateLeft(n % 8)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.rotateLeft(bitCount:
Int): UByte = toByte().rotateLeft(bitCount).toUByte()\n\n/**\n * Rotates the binary representation of this [UByte]
number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side
reenter the number as the most significant bits on the left side.\n * Rotating the number right by a negative bit
count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n
*\n * Rotating by a multiple of [UByte.SIZE_BITS] (8) returns the same number, or more generally\n *
`number.rotateRight(n) == number.rotateRight(n % 8)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.rotateRight(bitCount:
Int): UByte = toByte().rotateRight(bitCount).toUByte()\n\n/**\n * Counts the number of set bits in the binary
representation of this [UShort] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.countOneBits(): Int =
toUInt().countOneBits()\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the
binary representation of this [UShort] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.countLeadingZeroBits(): Int
= toShort().countLeadingZeroBits()\n\n/**\n * Counts the number of consecutive least significant bits that are zero
in the binary representation of this [UShort] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.countTrailingZeroBits(): Int
= toShort().countTrailingZeroBits()\n\n/**\n * Returns a number having a single bit set in the position of the most
significant set bit of this [UShort] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.takeHighestOneBit(): UShort
= toInt().takeHighestOneBit().toUShort()\n\n/**\n * Returns a number having a single bit set in the position of the
least significant set bit of this [UShort] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.takeLowestOneBit(): UShort
= toInt().takeLowestOneBit().toUShort()\n\n/**\n * Rotates the binary representation of this [UShort] number left
by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the
number as the least significant bits on the right side.\n * Rotating the number left by a negative bit count is the
same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n *
Rotating by a multiple of [UShort.SIZE_BITS] (16) returns the same number, or more generally\n *
`number.rotateLeft(n) == number.rotateLeft(n % 16)`\n

```



```

*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.rotateLeft(bitCount:
Int): UShort = toShort().rotateLeft(bitCount).toUShort()\n\n/**\n * Rotates the binary representation of this
[UShort] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the
right side reenter the number as the most significant bits on the left side.\n *\n * Rotating the number right by a
negative bit count is the same as rotating it left by the negated bit count.\n * `number.rotateRight(-n) ==
number.rotateLeft(n)`\n *\n * Rotating by a multiple of [UShort.SIZE_BITS] (16) returns the same number, or more
generally\n * `number.rotateRight(n) == number.rotateRight(n % 16)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.rotateRight(bitCount:
Int): UShort = toShort().rotateRight(bitCount).toUShort()\n", "\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n *\npackage kotlin.internal\n\n// (a - b) mod c\nprivate fun
differenceModulo(a: UInt, b: UInt, c: UInt): UInt {\n val ac = a % c\n val bc = b % c\n return if (ac >= bc) ac -
bc else ac - bc + c\n}\n\nprivate fun differenceModulo(a: ULong, b: ULong, c: ULong): ULong {\n val ac = a %
c\n val bc = b % c\n return if (ac >= bc) ac - bc else ac - bc + c\n}\n\n/**\n * Calculates the final element of a
bounded arithmetic progression, i.e. the last element of the progression which is in the range\n * from [start] to [end]
in case of a positive [step], or from [end] to [start] in case of a negative\n * [step].\n *\n * No validation on passed
parameters is performed. The given parameters should satisfy the condition:\n *\n * - either `step > 0` and `start <=
end`,\n * - or `step < 0` and `start >= end`.\n *\n * @param start first element of the progression\n * @param end
ending bound for the progression\n * @param step increment, or difference of successive elements in the
progression\n * @return the final element of the progression\n * @suppress\n
*\n@PublishedApi\n@SinceKotlin("1.3")\ninternal fun getProgressionLastElement(start: UInt, end: UInt, step:
Int): UInt = when {\n step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step.toUInt())\n
step < 0 -> if (start <= end) end else end + differenceModulo(start, end, (-step).toUInt())\n else -> throw
kotlin.IllegalArgumentException("Step is zero.")\n}\n\n/**\n * Calculates the final element of a bounded
arithmetic progression, i.e. the last element of the progression which is in the range\n * from [start] to [end] in case
of a positive [step], or from [end] to [start] in case of a negative\n * [step].\n *\n * No validation on passed
parameters is performed. The given parameters should satisfy the condition:\n *\n * - either `step > 0` and `start <=
end`,\n * - or `step < 0` and `start >= end`.\n *\n * @param start first element of the progression\n * @param end
ending bound for the progression\n * @param step increment, or difference of successive elements in the
progression\n * @return the final element of the progression\n * @suppress\n
*\n@PublishedApi\n@SinceKotlin("1.3")\ninternal fun getProgressionLastElement(start: ULong, end: ULong,
step: Long): ULong = when {\n step > 0 -> if (start >= end) end else end - differenceModulo(end, start,
step.toULong())\n step < 0 -> if (start <= end) end else end + differenceModulo(start, end, (-step).toULong())\n
else -> throw kotlin.IllegalArgumentException("Step is zero.")\n}\n", "\n * Copyright 2010-2021 JetBrains s.r.o.
and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n *\n@file:kotlin.jvm.JvmName("UStringsKt") // string
representation of unsigned numbers\n\npackage kotlin.text\n\n/**\n * Returns a string representation of this [Byte]
value in the specified [radix].\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for
number to string conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n//@kotlin.internal.InlineOnly\npublic
/*inline*/ fun UByte.toString(radix: Int): String = this.toInt().toString(radix)\n\n/**\n * Returns a string
representation of this [Short] value in the specified [radix].\n *\n * @throws IllegalArgumentException when [radix]
is not a valid radix for number to string conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n//@kotlin.internal.InlineOnly\npublic
/*inline*/ fun UShort.toString(radix: Int): String = this.toInt().toString(radix)\n\n\n/**\n * Returns a string
representation of this [Int] value in the specified [radix].\n *\n * @throws IllegalArgumentException when [radix] is

```

```

not a valid radix for number to string conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n//@kotlin.internal.InlineOnly\npublic /*inline*/ fun UInt.toString(radix: Int): String = this.toLong().toString(radix)\n\n**\n * Returns a string representation of this [Long] value in the specified [radix].\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun ULong.toString(radix: Int): String = ulongToString(this.toLong(), checkRadix(radix))\n\n**\n * Parses the string as a signed [UByte] number and returns the result.\n *\n * @throws NumberFormatException if the string is not a valid representation of a number.\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUByte(): UByte = toUByteOrNull() ?: numberFormatError(this)\n\n**\n * Parses the string as a signed [UByte] number and returns the result.\n *\n * @throws NumberFormatException if the string is not a valid representation of a number.\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUByte(radix: Int): UByte = toUByteOrNull(radix) ?: numberFormatError(this)\n\n**\n * Parses the string as a [UShort] number and returns the result.\n *\n * @throws NumberFormatException if the string is not a valid representation of a number.\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUShort(): UShort = toUShortOrNull() ?: numberFormatError(this)\n\n**\n * Parses the string as a [UShort] number and returns the result.\n *\n * @throws NumberFormatException if the string is not a valid representation of a number.\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUShort(radix: Int): UShort = toUShortOrNull(radix) ?: numberFormatError(this)\n\n**\n * Parses the string as an [UInt] number and returns the result.\n *\n * @throws NumberFormatException if the string is not a valid representation of a number.\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUInt(): UInt = toUIntOrNull() ?: numberFormatError(this)\n\n**\n * Parses the string as an [UInt] number and returns the result.\n *\n * @throws NumberFormatException if the string is not a valid representation of a number.\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUInt(radix: Int): UInt = toUIntOrNull(radix) ?: numberFormatError(this)\n\n**\n * Parses the string as a [ULong] number and returns the result.\n *\n * @throws NumberFormatException if the string is not a valid representation of a number.\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toULong(): ULong = toULongOrNull() ?: numberFormatError(this)\n\n**\n * Parses the string as a [ULong] number and returns the result.\n *\n * @throws NumberFormatException if the string is not a valid representation of a number.\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toULong(radix: Int): ULong = toULongOrNull(radix) ?: numberFormatError(this)\n\n**\n * Parses the string as an [UByte] number and returns the result\n *\n * or `null` if the string is not a valid representation of a number.\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUByteOrNull(): UByte? = toUByteOrNull(radix = 10)\n\n**\n * Parses the string as an [UByte] number and returns the result\n *\n * or `null` if the string is not a valid representation of a number.\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUByteOrNull(radix: Int): UByte? {\n val int = this.toUIntOrNull(radix) ?: return null\n if (int > UByte.MAX_VALUE) return null\n return int.toUByte()\n}\n\n**\n * Parses the string as an [UShort] number and returns the result\n *\n * or `null` if the string is not a valid representation of a number.\n

```

```

*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toUShortOrNull(): UShort? = toUShortOrNull(radix = 10)\n\n/**\n * Parses the string as an [UShort] number
and returns the result\n * or `null` if the string is not a valid representation of a number.\n *\n * @throws
IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toUShortOrNull(radix: Int): UShort? {\n val int = this.toUIntOrNull(radix) ?: return null\n if (int >
UShort.MAX_VALUE) return null\n return int.toUShort()\n}\n\n/**\n * Parses the string as an [UInt] number and
returns the result\n * or `null` if the string is not a valid representation of a number.\n *\n * @throws
IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toUIntOrNull(): UInt? = toUIntOrNull(radix = 10)\n\n/**\n * Parses the string as an [UInt] number and
returns the result\n * or `null` if the string is not a valid representation of a number.\n *\n * @throws
IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toUIntOrNull(radix: Int): UInt? {\n checkRadix(radix)\n\n val length = this.length\n if (length == 0)
return null\n\n val limit: UInt = UInt.MAX_VALUE\n val start: Int\n\n val firstChar = this[0]\n if (firstChar
< '0') {\n if (length == 1 || firstChar != '+') return null\n start = 1\n } else {\n start = 0\n }\n\n val
limitForMaxRadix = 119304647u // limit / 36\n\n var limitBeforeMul = limitForMaxRadix\n val uradix =
radix.toUInt()\n var result = 0u\n for (i in start until length) {\n val digit = digitOf(this[i], radix)\n\n if
(digit < 0) return null\n if (result > limitBeforeMul) {\n if (limitBeforeMul == limitForMaxRadix) {\n
 limitBeforeMul = limit / uradix\n\n if (result > limitBeforeMul) {\n return null\n
 }\n } else {\n return null\n }\n }\n\n result *= uradix\n\n val beforeAdding =
result\n result += digit.toUInt()\n if (result < beforeAdding) return null // overflow has happened\n }\n\n return result\n}\n\n/**\n * Parses the string as an [ULong] number and returns the result\n * or `null` if the string is
not a valid representation of a number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toULongOrNull(): ULong? = toULongOrNull(radix = 10)\n\n/**\n * Parses the string as an [ULong] number
and returns the result\n * or `null` if the string is not a valid representation of a number.\n *\n * @throws
IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toULongOrNull(radix: Int): ULong? {\n checkRadix(radix)\n\n val length = this.length\n if (length ==
0) return null\n\n val limit: ULong = ULong.MAX_VALUE\n val start: Int\n\n val firstChar = this[0]\n if
(firstChar < '0') {\n if (length == 1 || firstChar != '+') return null\n start = 1\n } else {\n start = 0\n
 }\n\n val limitForMaxRadix = 512409557603043100uL // limit / 36\n\n var limitBeforeMul =
limitForMaxRadix\n val uradix = radix.toULong()\n var result = 0uL\n for (i in start until length) {\n val
digit = digitOf(this[i], radix)\n\n if (digit < 0) return null\n if (result > limitBeforeMul) {\n if
(limitBeforeMul == limitForMaxRadix) {\n limitBeforeMul = limit / uradix\n\n if (result >
limitBeforeMul) {\n return null\n }\n } else {\n return null\n }\n
 }\n\n result *= uradix\n\n val beforeAdding = result\n result += digit.toUInt()\n if (result <
beforeAdding) return null // overflow has happened\n }\n\n return result\n}\n\n"/\n\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n@file:Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")\npackage kotlin\n\nimport
kotlin.annotation.AnnotationTarget.\n\nimport kotlin.internal.RequireKotlin\n\nimport
kotlin.internal.RequireKotlinVersionKind\n\n/**\n * Marks the API that is dependent on the experimental unsigned
types, including those types themselves.\n *\n * Usages of such API will be reported as warnings unless an explicit
opt-in with\n * the [OptIn] annotation, e.g. `@OptIn(ExperimentalUnsignedTypes::class)`\n * or with the `opt-
in=kotlin.ExperimentalUnsignedTypes` compiler option is given.\n *\n * It's recommended to propagate the
experimental status to the API that depends on unsigned types by annotating it with this annotation.\n

```

```

*\n@RequiresOptIn(level = RequiresOptIn.Level.WARNING)\n@MustBeDocumented\n@Target(CLASS,
ANNOTATION_CLASS, PROPERTY, FIELD, LOCAL_VARIABLE, VALUE_PARAMETER,
CONSTRUCTOR, FUNCTION, PROPERTY_GETTER, PROPERTY_SETTER,
TYPEALIAS)\n@Retention(AnnotationRetention.BINARY)\npublic annotation class
ExperimentalUnsignedTypes\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("MathKt")\n\npackage
kotlin.math\n\n// constants, can't use them from nativeMath as they are not constants there\n/** Ratio of the
circumference of a circle to its diameter, approximately 3.14159. *\n@SinceKotlin("1.2")\npublic const val PI:
Double = 3.141592653589793\n/** Base of the natural logarithms, approximately 2.71828.
*\n@SinceKotlin("1.2")\npublic const val E: Double = 2.718281828459045\n\n// region =====
Double Math =====\n/** Computes the sine of the angle [x]
given in radians.\n * \n * Special cases:\n * - `sin(NaN|+Inf|-Inf)` is `NaN`\n *\n@SinceKotlin("1.2")\npublic
expect fun sin(x: Double): Double\n/** Computes the cosine of the angle [x] given in radians.\n * \n * Special
cases:\n * - `cos(NaN|+Inf|-Inf)` is `NaN`\n *\n@SinceKotlin("1.2")\npublic expect fun cos(x: Double):
Double\n/** Computes the tangent of the angle [x] given in radians.\n * \n * Special cases:\n * - `tan(NaN|+Inf|-
Inf)` is `NaN`\n *\n@SinceKotlin("1.2")\npublic expect fun tan(x: Double): Double\n/** \n * Computes the arc
sine of the value [x];\n * the returned value is an angle in the range from `-PI/2` to `PI/2` radians.\n * \n * Special
cases:\n * - `asin(x)` is `NaN`, when `abs(x) > 1` or x is `NaN`\n *\n@SinceKotlin("1.2")\npublic expect fun
asin(x: Double): Double\n/** \n * Computes the arc cosine of the value [x];\n * the returned value is an angle in
the range from `0.0` to `PI` radians.\n * \n * Special cases:\n * - `acos(x)` is `NaN`, when `abs(x) > 1` or x is
`NaN`\n *\n@SinceKotlin("1.2")\npublic expect fun acos(x: Double): Double\n/** \n * Computes the arc tangent
of the value [x];\n * the returned value is an angle in the range from `-PI/2` to `PI/2` radians.\n * \n * Special cases:\n
* - `atan(NaN)` is `NaN`\n *\n@SinceKotlin("1.2")\npublic expect fun atan(x: Double): Double\n/** \n *
Returns the angle `theta` of the polar coordinates `(r, theta)` that correspond\n * to the rectangular coordinates `(x,
y)` by computing the arc tangent of the value [y] / [x];\n * the returned value is an angle in the range from `-PI` to
`PI` radians.\n * \n * Special cases:\n * - `atan2(0.0, 0.0)` is `0.0`\n * - `atan2(0.0, x)` is `0.0` for `x > 0` and `PI`
for `x < 0`\n * - `atan2(-0.0, x)` is `-0.0` for `x > 0` and `-PI` for `x < 0`\n * - `atan2(y, +Inf)` is `0.0` for `0 < y <
+Inf` and `-0.0` for `-Inf < y < 0`\n * - `atan2(y, -Inf)` is `PI` for `0 < y < +Inf` and `-PI` for `-Inf < y < 0`\n * -
`atan2(y, 0.0)` is `PI/2` for `y > 0` and `-PI/2` for `y < 0`\n * - `atan2(+Inf, x)` is `PI/2` for finite `x`\n * -
`atan2(-Inf, x)` is `-PI/2` for finite `x`\n * - `atan2(NaN, x)` and `atan2(y, NaN)` is `NaN`\n
*\n@SinceKotlin("1.2")\npublic expect fun atan2(y: Double, x: Double): Double\n/** \n * Computes the
hyperbolic sine of the value [x].\n * \n * Special cases:\n * - `sinh(NaN)` is `NaN`\n * - `sinh(+Inf)` is `+Inf`\n *
- `sinh(-Inf)` is `-Inf`\n *\n@SinceKotlin("1.2")\npublic expect fun sinh(x: Double): Double\n/** \n * Computes
the hyperbolic cosine of the value [x].\n * \n * Special cases:\n * - `cosh(NaN)` is `NaN`\n * - `cosh(+Inf|-Inf)` is
`+Inf`\n *\n@SinceKotlin("1.2")\npublic expect fun cosh(x: Double): Double\n/** \n * Computes the hyperbolic
tangent of the value [x].\n * \n * Special cases:\n * - `tanh(NaN)` is `NaN`\n * - `tanh(+Inf)` is `1.0`\n * - `tanh(-
Inf)` is `-1.0`\n *\n@SinceKotlin("1.2")\npublic expect fun tanh(x: Double): Double\n/** \n * Computes the
inverse hyperbolic sine of the value [x].\n * \n * The returned value is `y` such that `sinh(y) == x`.\n * \n * Special
cases:\n * - `asinh(NaN)` is `NaN`\n * - `asinh(+Inf)` is `+Inf`\n * - `asinh(-Inf)` is `-Inf`\n
*\n@SinceKotlin("1.2")\npublic expect fun asinh(x: Double): Double\n/** \n * Computes the inverse hyperbolic
cosine of the value [x].\n * \n * The returned value is positive `y` such that `cosh(y) == x`.\n * \n * Special cases:\n
* - `acosh(NaN)` is `NaN`\n * - `acosh(x)` is `NaN` when `x < 1`\n * - `acosh(+Inf)` is `+Inf`\n
*\n@SinceKotlin("1.2")\npublic expect fun acosh(x: Double): Double\n/** \n * Computes the inverse hyperbolic
tangent of the value [x].\n * \n * The returned value is `y` such that `tanh(y) == x`.\n * \n * Special cases:\n
* - `tanh(NaN)` is `NaN`\n * - `tanh(x)` is `NaN` when `x > 1` or `x < -1`\n * - `tanh(1.0)` is `+Inf`\n * - `tanh(-
1.0)` is `-Inf`\n *\n@SinceKotlin("1.2")\npublic expect fun atanh(x: Double): Double\n/** \n * Computes

```

`sqrt(x^2 + y^2)` without intermediate overflow or underflow.  
 \* Special cases:  
 - returns `+Inf` if any of arguments is infinite  
 - returns `NaN` if any of arguments is `NaN` and the other is not infinite

```

@SinceKotlin("1.2")
public expect fun hypot(x: Double, y: Double): Double
 * Computes the positive square root of the value [x].
 * Special cases:
 * - sqrt(x) is NaN when x < 0 or x is NaN
 *
@SinceKotlin("1.2")
public expect fun sqrt(x: Double): Double
 * Computes Euler's number e raised to the power of the value [x].
 * Special cases:
 * - exp(NaN) is NaN
 * - exp(+Inf) is +Inf
 * - exp(-Inf) is 0.0
 *
@SinceKotlin("1.2")
public expect fun exp(x: Double): Double
 * Computes exp(x) - 1.
 * This function can be implemented to produce more precise result for [x] near zero.
 * Special cases:
 * - expm1(NaN) is NaN
 * - expm1(+Inf) is +Inf
 * - expm1(-Inf) is -1.0
 * @see [exp] function.
 *
@SinceKotlin("1.2")
public expect fun expm1(x: Double): Double
 * Computes the logarithm of the value [x] to the given [base].
 * Special cases:
 * - log(x, b) is NaN if either x or b are NaN
 * - log(x, b) is NaN when x < 0 or b <= 0 or b == 1.0
 * - log(+Inf, +Inf) is NaN
 * - log(+Inf, b) is +Inf for b > 1 and -Inf for b < 1
 * - log(0.0, b) is -Inf for b > 1 and +Inf for b > 1
 * See also logarithm functions for common fixed bases: [ln], [log10] and [log2].
 *
@SinceKotlin("1.2")
public expect fun log(x: Double, base: Double): Double
 * Computes the natural logarithm (base E) of the value [x].
 * Special cases:
 * - ln(NaN) is NaN
 * - ln(x) is NaN when x < 0.0
 * - ln(+Inf) is +Inf
 * - ln(0.0) is -Inf
 *
@SinceKotlin("1.2")
public expect fun ln(x: Double): Double
 * Computes the common logarithm (base 10) of the value [x].
 * @see [ln] function for special cases.
 *
@SinceKotlin("1.2")
public expect fun log10(x: Double): Double
 * Computes the binary logarithm (base 2) of the value [x].
 * @see [ln] function for special cases.
 *
@SinceKotlin("1.2")
public expect fun log2(x: Double): Double
 * Computes ln(x + 1).
 * This function can be implemented to produce more precise result for [x] near zero.
 * Special cases:
 * - ln1p(NaN) is NaN
 * - ln1p(x) is NaN where x < -1.0
 * - ln1p(-1.0) is -Inf
 * - ln1p(+Inf) is +Inf
 * @see [ln] function
 * @see [expm1] function
 *
@SinceKotlin("1.2")
public expect fun ln1p(x: Double): Double
 * Rounds the given value [x] to an integer towards positive infinity.
 * @return the smallest double value that is greater than or equal to the given value [x] and is a mathematical integer.
 * Special cases:
 * - ceil(x) is x where x is NaN or +Inf or -Inf or already a mathematical integer.
 *
@SinceKotlin("1.2")
public expect fun ceil(x: Double): Double
 * Rounds the given value [x] to an integer towards negative infinity.
 * @return the largest double value that is smaller than or equal to the given value [x] and is a mathematical integer.
 * Special cases:
 * - floor(x) is x where x is NaN or +Inf or -Inf or already a mathematical integer.
 *
@SinceKotlin("1.2")
public expect fun floor(x: Double): Double
 * Rounds the given value [x] to an integer towards zero.
 * @return the value [x] having its fractional part truncated.
 * Special cases:
 * - truncate(x) is x where x is NaN or +Inf or -Inf or already a mathematical integer.
 *
@SinceKotlin("1.2")
public expect fun truncate(x: Double): Double
 * Rounds the given value [x] towards the closest integer with ties rounded towards even integer.
 * Special cases:
 * - round(x) is x where x is NaN or +Inf or -Inf or already a mathematical integer.
 *
@SinceKotlin("1.2")
public expect fun round(x: Double): Double
 * Returns the absolute value of the given value [x].
 * Special cases:
 * - abs(NaN) is NaN
 * @see absoluteValue extension property for [Double]
 *
@SinceKotlin("1.2")
public expect fun abs(x: Double): Double
 * Returns the sign of the given value [x]:
 * - -1.0 if the value is negative,
 * - zero if the value is zero,
 * - 1.0 if the value is positive
 * Special case:
 * - sign(NaN) is NaN
 *
@SinceKotlin("1.2")
public expect fun sign(x: Double): Double
 * Returns the smaller of two values.
 * If either value is NaN, then the result is NaN.
 *
@SinceKotlin("1.2")
public expect fun min(a: Double, b: Double): Double
 * Returns the greater of two values.
 * If either value is NaN, then the result is NaN.
 *
@SinceKotlin("1.2")
public expect fun max(a: Double, b: Double): Double
 * Returns the cube root of [x]. For any x, cbrt(-x) == -cbrt(x); that is, the cube root of a negative value is the negative of the cube root of that value's magnitude.
 * Special cases:
 * - If the argument is NaN, then the result is NaN.
 * - If the argument is infinite, then the result is an infinity with the same sign as the argument.

```

\* - If the argument is zero, then the result is a zero with the same sign as the argument.\n

\*\n@SinceKotlin("1.7")\n@ExperimentalStdlibApi\npublic expect fun cbrt(x: Double): Double\n\n//  
extensions\n\n/\*\*\n \* Raises this value to the power [x].\n \* \n \* Special cases:\n \* - `b.pow(0.0)` is `1.0`\n \* -  
`b.pow(1.0) == b`\n \* - `b.pow(NaN)` is `NaN`\n \* - `NaN.pow(x)` is `NaN` for `x != 0.0`\n \* - `b.pow(Inf)` is  
`NaN` for `abs(b) == 1.0`\n \* - `b.pow(x)` is `NaN` for `b < 0` and `x` is finite and not an integer\n

\*\n@SinceKotlin("1.2")\npublic expect fun Double.pow(x: Double): Double\n\n/\*\*\n \* Raises this value to the  
integer power [n].\n \* \n \* See the other overload of [pow] for details.\n \*\n@SinceKotlin("1.2")\npublic expect  
fun Double.pow(n: Int): Double\n\n/\*\*\n \* Returns the absolute value of this value.\n \* \n \* Special cases:\n \* -  
`NaN.absoluteValue` is `NaN`\n \* \n \* @see abs function\n \*\n@SinceKotlin("1.2")\npublic expect val  
Double.absoluteValue: Double\n\n/\*\*\n \* Returns the sign of this value:\n \* - `-1.0` if the value is negative,\n \* -  
zero if the value is zero,\n \* - `1.0` if the value is positive\n \* \n \* Special case:\n \* - `NaN.sign` is `NaN`\n

\*\n@SinceKotlin("1.2")\npublic expect val Double.sign: Double\n\n/\*\*\n \* Returns this value with the sign bit  
same as of the [sign] value.\n \* \n \* If [sign] is `NaN` the sign of the result is undefined.\n

\*\n@SinceKotlin("1.2")\npublic expect fun Double.withSign(sign: Double): Double\n\n/\*\*\n \* Returns this value  
with the sign bit same as of the [sign] value.\n \*\n@SinceKotlin("1.2")\npublic expect fun Double.withSign(sign:  
Int): Double\n\n/\*\*\n \* Returns the ulp (unit in the last place) of this value.\n \* \n \* An ulp is a positive distance  
between this value and the next nearest [Double] value larger in magnitude.\n \* \n \* Special Cases:\n \* - `NaN.ulp`  
is `NaN`\n \* - `x.ulp` is `+Inf` when `x` is `+Inf` or `-Inf`\n \* - `0.0.ulp` is `Double.MIN\_VALUE`\n

\*\n@SinceKotlin("1.2")\npublic expect val Double.ulp: Double\n\n/\*\*\n \* Returns the [Double] value nearest to  
this value in direction of positive infinity.\n \*\n@SinceKotlin("1.2")\npublic expect fun Double.nextUp():  
Double\n\n/\*\*\n \* Returns the [Double] value nearest to this value in direction of negative infinity.\n

\*\n@SinceKotlin("1.2")\npublic expect fun Double.nextDown(): Double\n\n/\*\*\n \* Returns the [Double] value  
nearest to this value in direction from this value towards the value [to].\n \* \n \* Special cases:\n \* -  
`x.nextTowards(y)` is `NaN` if either `x` or `y` are `NaN`\n \* - `x.nextTowards(x) == x`\n \* \n

\*\n@SinceKotlin("1.2")\npublic expect fun Double.nextTowards(to: Double): Double\n\n/\*\*\n \* Rounds this  
[Double] value to the nearest integer and converts the result to [Int].\n \* \n \* Ties are rounded towards positive infinity.\n \* \n \* Special cases:\n \* - `x.roundToInt() == Int.MAX\_VALUE` when `x > Int.MAX\_VALUE`\n \* -  
`x.roundToInt() == Int.MIN\_VALUE` when `x < Int.MIN\_VALUE`\n \* \n \* @throws IllegalArgumentException  
when this value is `NaN`\n \*\n@SinceKotlin("1.2")\npublic expect fun Double.roundToInt(): Int\n\n/\*\*\n \*  
Rounds this [Double] value to the nearest integer and converts the result to [Long].\n \* \n \* Ties are rounded towards  
positive infinity.\n \* \n \* Special cases:\n \* - `x.roundToLong() == Long.MAX\_VALUE` when `x >  
Long.MAX\_VALUE`\n \* - `x.roundToLong() == Long.MIN\_VALUE` when `x < Long.MIN\_VALUE`\n \* \n \*  
@throws IllegalArgumentException when this value is `NaN`\n \*\n@SinceKotlin("1.2")\npublic expect fun  
Double.roundToLong(): Long\n\n// endregion\n\n\n// region ===== Float Math  
===== \n\n/\*\*\n \* Computes the sine of the angle [x] given in  
radians.\n \* \n \* Special cases:\n \* - `sin(NaN|+Inf|-Inf)` is `NaN`\n \*\n@SinceKotlin("1.2")\npublic expect fun  
sin(x: Float): Float\n\n/\*\*\n \* Computes the cosine of the angle [x] given in radians.\n \* \n \* Special cases:\n \* -  
`cos(NaN|+Inf|-Inf)` is `NaN`\n \*\n@SinceKotlin("1.2")\npublic expect fun cos(x: Float): Float\n\n/\*\*\n \* Computes  
the tangent of the angle [x] given in radians.\n \* \n \* Special cases:\n \* - `tan(NaN|+Inf|-Inf)` is `NaN`\n

\*\n@SinceKotlin("1.2")\npublic expect fun tan(x: Float): Float\n\n/\*\*\n \* Computes the arc sine of the value  
[x];\n \* \n \* the returned value is an angle in the range from  $-\pi/2$  to  $\pi/2$  radians.\n \* \n \* Special cases:\n \* -  
`asin(x)` is `NaN`, when `abs(x) > 1` or x is `NaN`\n \*\n@SinceKotlin("1.2")\npublic expect fun asin(x: Float):  
Float\n\n/\*\*\n \* Computes the arc cosine of the value [x];\n \* \n \* the returned value is an angle in the range from `0.0`  
to  $\pi$  radians.\n \* \n \* Special cases:\n \* - `acos(x)` is `NaN`, when `abs(x) > 1` or x is `NaN`\n

\*\n@SinceKotlin("1.2")\npublic expect fun acos(x: Float): Float\n\n/\*\*\n \* Computes the arc tangent of the value  
[x];\n \* \n \* the returned value is an angle in the range from  $-\pi/2$  to  $\pi/2$  radians.\n \* \n \* Special cases:\n \* -  
`atan(NaN)` is `NaN`\n \*\n@SinceKotlin("1.2")\npublic expect fun atan(x: Float): Float\n\n/\*\*\n \* Returns the  
angle `theta` of the polar coordinates `(r, theta)` that correspond\n \* to the rectangular coordinates `(x, y)` by

computing the arc tangent of the value  $y / x$ ; the returned value is an angle in the range from  $-\pi$  to  $\pi$  radians.

**Special cases:**

- $\text{atan2}(0.0, 0.0)$  is  $0.0$
- $\text{atan2}(0.0, x)$  is  $0.0$  for  $x > 0$  and  $\pi$  for  $x < 0$
- $\text{atan2}(-0.0, x)$  is  $-0.0$  for  $x > 0$  and  $-\pi$  for  $x < 0$
- $\text{atan2}(y, +\text{Inf})$  is  $0.0$  for  $0 < y < +\text{Inf}$  and  $-0.0$  for  $-\text{Inf} < y < 0$
- $\text{atan2}(y, -\text{Inf})$  is  $\pi$  for  $0 < y < +\text{Inf}$  and  $-\pi$  for  $-\text{Inf} < y < 0$
- $\text{atan2}(y, 0.0)$  is  $\pi/2$  for  $y > 0$  and  $-\pi/2$  for  $y < 0$
- $\text{atan2}(+\text{Inf}, x)$  is  $\pi/2$  for finite  $x$
- $\text{atan2}(-\text{Inf}, x)$  is  $-\pi/2$  for finite  $x$
- $\text{atan2}(\text{NaN}, x)$  and  $\text{atan2}(y, \text{NaN})$  is  $\text{NaN}$

```

*\/n@SinceKotlin("1.2")\npublic expect fun atan2(y: Float, x: Float): Float\n\n**\n * Computes the hyperbolic
sine of the value [x].\n * Special cases:\n * - sinh(NaN) is NaN\n * - sinh(+Inf) is +Inf\n * - sinh(-
Inf) is -Inf\n
*\/n@SinceKotlin("1.2")\npublic expect fun sinh(x: Float): Float\n\n**\n * Computes the
hyperbolic cosine of the value [x].\n * Special cases:\n * - cosh(NaN) is NaN\n * - cosh(+Inf|-Inf) is
+Inf\n
*\/n@SinceKotlin("1.2")\npublic expect fun cosh(x: Float): Float\n\n**\n * Computes the hyperbolic
tangent of the value [x].\n * Special cases:\n * - tanh(NaN) is NaN\n * - tanh(+Inf) is 1.0\n * - tanh(-
Inf) is -1.0\n
*\/n@SinceKotlin("1.2")\npublic expect fun tanh(x: Float): Float\n\n**\n * Computes the inverse
hyperbolic sine of the value [x].\n * The returned value is `y` such that `sinh(y) == x`.\n * Special cases:\n
* - asinh(NaN) is NaN\n * - asinh(+Inf) is +Inf\n * - asinh(-Inf) is -Inf\n
*\/n@SinceKotlin("1.2")\npublic expect fun asinh(x: Float): Float\n\n**\n * Computes the inverse hyperbolic
cosine of the value [x].\n * The returned value is positive `y` such that `cosh(y) == x`.\n * Special cases:\n
* - acosh(NaN) is NaN\n * - acosh(x) is NaN when `x < 1`\n * - acosh(+Inf) is +Inf\n
*\/n@SinceKotlin("1.2")\npublic expect fun acosh(x: Float): Float\n\n**\n * Computes the inverse hyperbolic
tangent of the value [x].\n * The returned value is `y` such that `tanh(y) == x`.\n * Special cases:\n
* - tanh(NaN) is NaN\n * - tanh(x) is NaN when `x > 1` or `x < -1`\n * - tanh(1.0) is +Inf\n * - tanh(-
1.0) is -Inf\n
*\/n@SinceKotlin("1.2")\npublic expect fun atanh(x: Float): Float\n\n**\n * Computes `sqrt(x^2 +
y^2)` without intermediate overflow or underflow.\n * Special cases:\n * - returns +Inf if any of arguments is
infinite\n * - returns NaN if any of arguments is NaN and the other is not infinite\n
*\/n@SinceKotlin("1.2")\npublic expect fun hypot(x: Float, y: Float): Float\n\n**\n * Computes the positive
square root of the value [x].\n * Special cases:\n * - sqrt(x) is NaN when `x < 0` or `x` is NaN\n
*\/n@SinceKotlin("1.2")\npublic expect fun sqrt(x: Float): Float\n\n**\n * Computes Euler's number `e` raised to
the power of the value [x].\n * Special cases:\n * - exp(NaN) is NaN\n * - exp(+Inf) is +Inf\n * -
exp(-Inf) is 0.0\n
*\/n@SinceKotlin("1.2")\npublic expect fun exp(x: Float): Float\n\n**\n * Computes `exp(x)
- 1`.\n * This function can be implemented to produce more precise result for [x] near zero.\n * Special
cases:\n * - expm1(NaN) is NaN\n * - expm1(+Inf) is +Inf\n * - expm1(-Inf) is -1.0\n * @see
[exp] function.\n
*\/n@SinceKotlin("1.2")\npublic expect fun expm1(x: Float): Float\n\n**\n * Computes the
logarithm of the value [x] to the given [base].\n * Special cases:\n * - log(x, b) is NaN if either `x` or `b` are
NaN\n * - log(x, b) is NaN when `x < 0` or `b <= 0` or `b == 1.0`\n * - log(+Inf, +Inf) is NaN\n * -
log(+Inf, b) is +Inf for `b > 1` and -Inf for `b < 1`\n * - log(0.0, b) is -Inf for `b > 1` and +Inf for `b > 1`\n
*\/n * See also logarithm functions for common fixed bases: [ln], [log10] and [log2].\n
*\/n@SinceKotlin("1.2")\npublic expect fun log(x: Float, base: Float): Float\n\n**\n * Computes the natural
logarithm (base 'E') of the value [x].\n * Special cases:\n * - ln(NaN) is NaN\n * - ln(x) is NaN when
`x < 0.0`\n * - ln(+Inf) is +Inf\n * - ln(0.0) is -Inf\n
*\/n@SinceKotlin("1.2")\npublic expect fun ln(x:
Float): Float\n\n**\n * Computes the common logarithm (base 10) of the value [x].\n * @see [ln] function for
special cases.\n
*\/n@SinceKotlin("1.2")\npublic expect fun log10(x: Float): Float\n\n**\n * Computes the binary
logarithm (base 2) of the value [x].\n * @see [ln] function for special cases.\n
*\/n@SinceKotlin("1.2")\npublic
expect fun log2(x: Float): Float\n\n**\n * Computes `ln(a + 1)`.\n * This function can be implemented to
produce more precise result for [x] near zero.\n * Special cases:\n * - ln1p(NaN) is NaN\n * - ln1p(x) is
NaN where `x < -1.0`\n * - ln1p(-1.0) is -Inf\n * - ln1p(+Inf) is +Inf\n * @see [ln] function\n * @see
[expm1] function\n
*\/n@SinceKotlin("1.2")\npublic expect fun ln1p(x: Float): Float\n\n**\n * Rounds the given
value [x] to an integer towards positive infinity.\n * @return the smallest Float value that is greater than or equal
to the given value [x] and is a mathematical integer.\n * Special cases:\n * - ceil(x) is `x` where `x` is NaN

```

or `+Inf` or `-Inf` or already a mathematical integer.

```

@SinceKotlin("1.2")
public expect fun ceil(x: Float):
Float
Rounds the given value [x] to an integer towards negative infinity.
@return the largest Float value that is smaller than or equal to the given value [x] and is a mathematical integer.
Special cases:
- floor(x) is x where x is NaN or +Inf or -Inf or already a mathematical integer.

```

```

@SinceKotlin("1.2")
public expect fun floor(x: Float): Float
Rounds the given value [x] to an integer towards zero.
@return the value [x] having its fractional part truncated.
Special cases:
- truncate(x) is x where x is NaN or +Inf or -Inf or already a mathematical integer.

```

```

@SinceKotlin("1.2")
public expect fun truncate(x: Float): Float
Rounds the given value [x] towards the closest integer with ties rounded towards even integer.
Special cases:
- round(x) is x where x is NaN or +Inf or -Inf or already a mathematical integer.

```

```

@SinceKotlin("1.2")
public expect fun round(x: Float): Float
Returns the absolute value of the given value [x].
Special cases:
- abs(NaN) is NaN
@see absoluteValue extension property for [Float]

```

```

@SinceKotlin("1.2")
public expect fun abs(x: Float): Float
Returns the sign of the given value [x]:
- -1.0 if the value is negative,
- zero if the value is zero,
- 1.0 if the value is positive
Special case:
- sign(NaN) is NaN

```

```

@SinceKotlin("1.2")
public expect fun sign(x: Float): Float
Returns the smaller of two values.
If either value is NaN, then the result is NaN.

```

```

@SinceKotlin("1.2")
public expect fun min(a: Float, b: Float): Float
Returns the greater of two values.
If either value is NaN, then the result is NaN.

```

```

@SinceKotlin("1.2")
public expect fun max(a: Float, b: Float): Float
Returns the cube root of [x]. For any x, cbrt(-x) == -cbrt(x); that is, the cube root of a negative value is the negative of the cube root of that value's magnitude.
Special cases:
- If the argument is NaN, then the result is NaN.
- If the argument is infinite, then the result is an infinity with the same sign as the argument.
- If the argument is zero, then the result is a zero with the same sign as the argument.

```

```

@SinceKotlin("1.7")
@ExperimentalStdlibApi
public expect fun cbrt(x: Float): Float // extensions
Raises this value to the power [x].
Special cases:
- b.pow(0.0) is 1.0
- b.pow(1.0) == b
- b.pow(NaN) is NaN
- NaN.pow(x) is NaN for x != 0.0
- b.pow(Inf) is NaN for abs(b) == 1.0
- b.pow(x) is NaN for b < 0 and x is finite and not an integer

```

```

@SinceKotlin("1.2")
public expect fun Float.pow(x: Float): Float
Raises this value to the integer power [n].
See the other overload of [pow] for details.

```

```

@SinceKotlin("1.2")
public expect fun Float.pow(n: Int): Float
Returns the absolute value of this value.
Special cases:
- NaN.absoluteValue is NaN
@see abs function

```

```

@SinceKotlin("1.2")
public expect val Float.absoluteValue: Float
Returns the sign of this value:
- -1.0 if the value is negative,
- zero if the value is zero,
- 1.0 if the value is positive
Special case:
- NaN.sign is NaN

```

```

@SinceKotlin("1.2")
public expect val Float.sign: Float
Returns this value with the sign bit same as of the [sign] value.
If [sign] is NaN the sign of the result is undefined.

```

```

@SinceKotlin("1.2")
public expect fun Float.withSign(sign: Float): Float
Returns this value with the sign bit same as of the [sign] value.

```

```

@SinceKotlin("1.2")
public expect fun Float.withSign(sign: Int): Float
Rounds this [Float] value to the nearest integer and converts the result to [Int].
Ties are rounded towards positive infinity.
Special cases:
- x.roundToInt() == Int.MAX_VALUE when x > Int.MAX_VALUE
- x.roundToInt() == Int.MIN_VALUE when x < Int.MIN_VALUE
@throws IllegalArgumentException when this value is NaN

```

```

@SinceKotlin("1.2")
public expect fun Float.roundToInt(): Int
Rounds this [Float] value to the nearest integer and converts the result to [Long].
Ties are rounded towards positive infinity.
Special cases:
- x.roundToLong() == Long.MAX_VALUE when x > Long.MAX_VALUE
- x.roundToLong() == Long.MIN_VALUE when x < Long.MIN_VALUE
@throws IllegalArgumentException when this value is NaN

```

```

@SinceKotlin("1.2")
public expect fun Float.roundToLong(): Long // endregion
===== Integer Math =====
Returns the absolute value of the given value [n].
Special cases:
- abs(Int.MIN_VALUE) is Int.MIN_VALUE due to an overflow
@see absoluteValue extension property for [Int]

```



```

*\n@SinceKotlin("1.2")\npublic expect fun abs(n: Int): Int\n\n**\n * Returns the smaller of two values.\n
*\n@SinceKotlin("1.2")\npublic expect fun min(a: Int, b: Int): Int\n\n**\n * Returns the greater of two values.\n
*\n@SinceKotlin("1.2")\npublic expect fun max(a: Int, b: Int): Int\n\n**\n * Returns the absolute value of this
value.\n * \n * Special cases:\n * - `Int.MIN_VALUE.absoluteValue` is `Int.MIN_VALUE` due to an overflow\n
*\n * @see abs function\n
*\n@SinceKotlin("1.2")\npublic expect val Int.absoluteValue: Int\n\n**\n * Returns
the sign of this value:\n * - `-1` if the value is negative,\n * - `0` if the value is zero,\n * - `1` if the value is
positive\n
*\n@SinceKotlin("1.2")\npublic expect val Int.sign: Int\n\n\n**\n * Returns the absolute value of the
given value [n].\n * \n * Special cases:\n * - `abs(Long.MIN_VALUE)` is `Long.MIN_VALUE` due to an
overflow\n * \n * @see absoluteValue extension property for [Long]\n
*\n@SinceKotlin("1.2")\npublic expect fun
abs(n: Long): Long\n\n**\n * Returns the smaller of two values.\n
*\n@SinceKotlin("1.2")\npublic expect fun
min(a: Long, b: Long): Long\n\n**\n * Returns the greater of two values.\n
*\n@SinceKotlin("1.2")\npublic
expect fun max(a: Long, b: Long): Long\n\n**\n * Returns the absolute value of this value.\n * \n * Special cases:\n
* - `Long.MIN_VALUE.absoluteValue` is `Long.MIN_VALUE` due to an overflow\n * \n * @see abs function\n
*\n@SinceKotlin("1.2")\npublic expect val Long.absoluteValue: Long\n\n**\n * Returns the sign of this value:\n
* - `-1` if the value is negative,\n * - `0` if the value is zero,\n * - `1` if the value is positive\n
\n@SinceKotlin("1.2")\npublic expect val Long.sign: Int\n\n\n// endregion\n", "/\n * Copyright 2010-2022
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\n**\n * Exposes
the JavaScript [Math object](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/Math)
to Kotlin.\n
*\n@PublishedApi\n@JsName("Math")\ninternal external object JsMath {\n val LN2: Double\n
fun abs(value: Double): Double\n fun acos(value: Double): Double\n fun asin(value: Double): Double\n fun
atan(value: Double): Double\n fun atan2(y: Double, x: Double): Double\n fun cos(value: Double): Double\n
fun sin(value: Double): Double\n fun exp(value: Double): Double\n fun max(vararg values: Int): Int\n fun
max(vararg values: Float): Float\n fun max(vararg values: Double): Double\n fun min(vararg values: Int): Int\n
fun min(vararg values: Float): Float\n fun min(vararg values: Double): Double\n fun sqrt(value: Double):
Double\n fun tan(value: Double): Double\n fun log(value: Double): Double\n fun cbrt(value: Double):
Double\n fun pow(base: Double, exp: Double): Double\n fun round(value: Number): Double\n fun floor(value:
Number): Double\n fun ceil(value: Number): Double\n}\n\ninternal const val defineTaylorNBound = "\\\"\n var
epsilon = 2.220446049250313E-16;\n var taylor_2_bound = Math.sqrt(epsilon);\n var taylor_n_bound =
Math.sqrt(taylor_2_bound);\n\\\"\n\ninternal const val defineUpperTaylor2Bound = "\\\"\n
$defineTaylorNBound\n var upper_taylor_2_bound = 1/taylor_2_bound;\n\\\"\n\ninternal const val
defineUpperTaylorNBound = "\\\"\n $defineUpperTaylor2Bound\n var upper_taylor_n_bound =
1/taylor_n_bound;\n\\\"\n\n", "names": [], "mappings": "AAWC,CAXA,yB;EACG,IAAI,OAAO,MAAO,KAAI,UAA
W,IAAG,MAAM,IAAIC,C;IACI,MAAM,CAAC,QAAD,EAAW,CAAC,SAAD,CAAX,EAAwB,OAAxB,C;SAEL
,IAAI,OAAO,OAAQ,KAAI,QAAvB,C;IACD,OAAO,CAAC,MAAM,QAAP,C;;IAGP,IAAI,OAAQ,GAAE,E;IAC
d,OAAO,CAAC,IAAI,OAAL,C;;CAEd,CAAC,IAAD,EAAO,kB;EACJ,IAAI,IAAI,M;ECPU;;;IAAtB,MAAM,eAA
gB,GAAE,a;IACpB,OAAoD,CAA5C,KAAK,QAAQ,CAAC,CAAD,CAAI,IAAG,CAAe,YAAW,SAAW,KAAgC,
AAC,OAAQ,KAAI,c;G;EAGxE,MAAM,YAAa,GAAE,a;IACjB,OAAO,CAAe,YAAW,SAAU,IAAG,CAAC,OAA
Q,KAAI,c;G;EAGID,MAAM,aAAc,GAAE,a;IACIB,OAAO,CAAe,YAAW,U;G;EAGxB,MAAM,YAAa,GAAE,a;
IACjB,OAAO,CAAe,YAAW,WAAY,IAAG,CAAC,OAAQ,KAAI,W;G;EAGpD,MAAM,WAAY,GAAE,a;IACbB
,OAAO,CAAe,YAAW,U;G;EAGxB,MAAM,aAAc,GAAE,a;IACIB,OAAO,CAAe,YAAW,Y;G;EAGxB,MAAM,c
AAe,GAAE,a;IACnB,OAAO,CAAe,YAAW,Y;G;EAGxB,MAAM,YAAa,GAAE,a;IACjB,OAAO,KAAK,QAAQ,
CAAC,CAAD,CAAI,IAAG,CAAC,OAAQ,KAAI,W;G;EAG5C,MAAM,QAAS,GAAE,a;IACb,OAAO,KAAK,QA
AQ,CAAC,CAAD,CAAI,IAAG,CAAC,CAAC,O;G;EAGjC,MAAM,WAAY,GAAE,a;IACbB,OAAO,KAAK,QA
AQ,CAAC,CAAD,CAAI,IAAG,WAAW,OAAO,CAAC,CAAD,C;G;EAGjD,MAAM,cAAe,GAAE,a;IACnB,IAAI,C
AAE,KAAI,IAAV,C;MAAgB,OAAO,M;IACvB,IAAI,WAAW,MAAM,YAAY,CAAC,CAAD,CAAI,GAAE,MAA
M,aAAR,GAAwB,MAAM,S;IACnE,OAAO,GAAl,GAAE,KAAK,UAAU,IAAI,KAAK,CAAC,CAAD,EAAl,a;M
AAc,OAAO,QAAQ,CAAC,CAAD,C;KAAjC,CAAwC,KAAK,CAAC,IAAD,CAAO,GAAE,G;G;EAGF,MAAM,k

```

BAAmB,GAAE,e;IACvB,OAAO,MAAM,OAAO,YAAY,wBAAwB,CAAC,GAAD,C;G;EAG5D,MAAM,YAAa,G  
AAE,gB;IACjB,IAAI,CAAE,KAAI,CAAV,C;MACI,OAAO,I;IAEX,IAAI,CAAE,KAAI,IAAK,IAAG,CAAE,KA  
AI,IAAK,IAAG,CAAC,MAAM,WAAW,CAAC,CAAD,CAAI,IAAG,CAAC,OAAQ,KAAI,CAAC,OAAvE,C;MA  
CI,OAAO,K;IAGX,KAAK,IAAI,IAAI,CAAR,EA AW,IAAI,CAAC,OAArB,EAA8B,CAAE,GAAE,CAAIC,EAAq  
C,CAAC,EAAtC,C;MACI,IAAI,CAAC,MAAM,OAAO,CAAC,CAAC,CAAC,CAAD,CAAF,EAAO,CAAC,CAA  
C,CAAD,CAAR,CAAIB,C;QACI,OAAO,K;IAGf,OAAO,I;G;EAGX,MAAM,gBAaIB,GAAE,gB;IACrB,OAAO,  
MAAM,OAAO,YAAY,sBAAsB,CAAC,CAAD,EA AI,CAAJ,C;G;EAG1D,MAAM,cAAe,GAAE,e;IACnB,IAAI,G  
AAI,KAAI,IAAZ,C;MAAkB,OAAO,C;IACzB,IAAI,SAAS,C;IACb,KAAK,IAAI,IAAI,CAAR,EA AW,IAAI,GAA  
G,OAAvB,EAAgC,CAAE,GAAE,CAApC,EAAuC,CAAC,EAAX,C;MACI,MAAO,GAAqB,CAAjB,EAAG,GAA  
E,MAAO,GAAE,CAAG,IAAE,MAAM,SAAS,CAAC,GAAG,CAAC,CAAD,CAAJ,CAAU,GAAE,C;IAE7D,OA  
AO,M;G;EAGX,MAAM,kBAAmB,GAAE,e;IACvB,OAAO,MAAM,OAAO,YAAY,wBAAwB,CAAC,GAAD,C;G  
;EAG5D,MAAM,mBAAoB,GAAE,iB;IACxB,KAAK,KAAK,CAAC,MAAM,gBAAP,C;G;ECPfQ;IAAtB,MAA  
M,eAAgB,GAAE,mB;IACpB,CAAC,aAAc,GAAE,I;IACjB,OAAO,C;G;EAGX,MAAM,uBAAwB,GAAE,4C;IAC  
5B,MAAM,IAAK,GAAE,M;IACb,MAAM,IAAK,GAAE,M;IACb,MAAM,aAAc,GAAE,I;IACtB,OAAO,mBAAm  
B,CAAC,MAAD,EAAS,MAAT,EAiB,6BAA6B,CAAC,UAAD,CAA9C,C;G;EAG9B,iD;IACI,GAAG,WAA Y,G  
AAE,sBAAsB,CAAC,OAAO,MAAO,KAAI,UAAW,GAAE,KAAK,QAAP,GAakB,KAAK,UArD,C;IACvC,GA  
AG,YAAa,GAAE,G;IACIB,OAAO,G;G;EAGX,IAAI,gCAAqC,CACc,CACI,OADJ,EACa,CAAE,KAAF,EAAS,  
IAAT,EAAe,oBAaf,EAAqC,Y;IAC1C,OAAO,MAAM,OAAO,QAAQ,kB;GADvB,CADb,EAI,SAJJ,EAIE,CAAE  
,KAAF,EAAS,IAAT,EAAe,oBAaf,EAAqC,Y;IAC5C,OAAO,MAAM,OAAO,QAAQ,W;GADrB,CAJf,CADgC,E  
AShC,CACI,OADJ,EACa,CAAE,KAAF,EAAS,IAAT,EAAe,oBAaf,EAAqC,Y;IAC1C,OAAO,MAAM,OAAO,Q  
AAQ,kB;GADvB,CADb,EAI,SAJJ,EAIE,CAAE,KAAF,EAAS,IAAT,EAAe,oBAaf,EAAqC,Y;IAC5C,OAAO,M  
AAM,OAAO,QAAQ,W;GADrB,CAJf,CATgC,C;EAmBpC,uC;IACI,IAAI,KAAK,MAAO,KAAI,IAApB,C;MACI,  
KAAK,MAAO,GAAE,CACV,UADU,EACE,CAAC,KAAK,qBAAqB,EAA3B,CADF,EA EV,SAFU,EAEC,IAFD,  
EAGV,SAHU,EAGC,EAHD,EAIV,UAJU,EAIE,EAJF,EAKV,KALU,EAKH,EALG,EAMV,aANU,EAMK,EANL,  
C;IASIB,OAAO,KAAK,M;G;EChDD;IAAf,MAAM,QAAS,GAAE,a;IACb,OAAoB,CAAZ,CAAE,GAAE,KAA  
Q,KAAG,EAAG,IAAG,E;G;EAGjC,MAAM,OAAQ,GAAE,a;IACZ,OAAkB,CAAV,CAAE,GAAE,GAAM,KAA  
G,EAAG,IAAG,E;G;EAG/B,MAAM,OAAQ,GAAE,a;IACZ,OAAO,CAAE,GAAE,K;G;EAGf,MAAM,aAAc,GA  
AE,a;IACIB,OAAO,CAAE,YAAW,MAAM,KAAM,GAAE,CAAF,GAAM,MAAM,KAAK,WAAW,CAAC,CAAD  
,C;G;EAGhE,MAAM,YAAa,GAAE,a;IACjB,OAAO,CAAE,YAAW,MAAM,KAAM,GAAE,CAAC,MAAM,EA  
T,GAAc,MAAM,YAAY,CAAC,CAAD,C;G;EAGpE,MAAM,cAAe,GAAE,a;IACnB,OAAO,MAAM,QAAQ,CAA  
C,MAAM,YAAY,CAAC,CAAD,CAAnB,C;G;EAGzB,MAAM,aAAc,GAAE,a;IACIB,OAAO,MAAM,OAAO,CA  
AC,MAAM,YAAY,CAAC,CAAD,CAAnB,C;G;EAGxB,MAAM,eAAgB,GAAE,a;IACpB,OAAO,CAAC,C;G;EA  
GZ,MAAM,aAAc,GAAE,a;IACIB,OAAO,MAAM,OAAO,CAAC,MAAM,YAAY,CAAC,CAAD,CAAnB,C;G;EA  
GxB,MAAM,YAAa,GAAE,a;IACjB,IAAI,CAAE,GAAE,UAR,C;MAAoB,OAAO,U;IAC3B,IAAI,CAAE,GAAE  
,WAAR,C;MAAqB,OAAO,W;IAC5B,OAAO,CAAE,GAAE,C;G;EAGf,MAAM,YAAa,GAAE,a;IACjB,IAAI,CA  
AE,IAAG,IAAT,C;MAAe,OAAO,C;IACtB,IAAI,CAAE,YAAW,MAAM,UAAvB,C;MAAmC,OAAO,C;IAC1C,O  
AAO,IAAI,MAAM,UAAV,CAAqB,CAArB,C;G;EAGX,MAAM,UAAW,GAAE,a;IACf,IAAI,CAAE,IAAG,IAAT  
,C;MAAe,OAAO,C;IACtB,OAAO,MAAM,OAAO,CAAC,CAAD,C;G;ECIDV;IAAd,MAAM,OAAQ,GAAE,sB;I  
ACZ,IAAI,IAAK,IAAG,IAAZ,C;MACI,OAAO,IAAK,IAAG,I;IAGnB,IAAI,IAAK,IAAG,IAAZ,C;MACI,OAAO,  
K;IAGX,IAAI,IAAK,KAAI,IAAb,C;MACI,OAAO,IAAK,KAAI,I;IAGpB,IAAI,OAAO,IAAK,KAAI,QAAS,IAA  
G,OAAO,IAAI,OAAQ,KAAI,UAAvD,C;MACI,OAAO,IAAI,OAAO,CAAC,IAAD,C;IAGtB,IAAI,OAAO,IAAK,  
KAAI,QAAS,IAAG,OAAO,IAAK,KAAI,QAAd,C;MACI,OAAO,IAAK,KAAI,IAAK,KAAI,IAAK,KAAI,CAA  
E,IAAG,CAAE,GAAE,IAAK,KAAI,CAAE,GAAE,IAAnC,C;IAGzB,OAAO,IAAK,KAAI,I;G;EAGpB,MAAM,S  
AAU,GAAE,e;IACd,IAAI,GAAL,IAAG,IAAX,C;MACI,OAAO,C;IAEX,IAAI,UAAU,OAAO,G;IACrB,IAAI,QA  
AS,KAAI,OAAjB,C;MACI,OAAO,UAAW,KAAI,OAAO,GAAG,SAAU,GAAE,GAAG,SAAS,EAAd,GAAmB,iB  
AAiB,CAAC,GAAD,C;IAEIF,IAAI,UAAW,KAAI,OAAAnB,C;MACI,OAAO,iBAaiB,CAAC,GAAD,C;IAE5B,I  
AAI,QAAS,KAAI,OAAjB,C;MACI,OAAO,MAAM,eAAe,CAAC,GAAD,C;IAEhC,IAAI,SAAU,KAAI,OAAIB,C  
;MACI,OAAO,MAAM,CAAC,GAAD,C;IAGjB,IAAI,MAAM,MAAM,CAAC,GAAD,C;IACbB,OAAO,iBAaiB,



,Y;IAC/B,OAAO,IAAI,MAAO,GAAE,MAAM,KAAK,gBAaIB,GACzC,IAAI,mBAAmB,E;GAFD,yD;EAM/B,M  
AAM,KAAK,UAAU,SAAU,GAAE,Y;IAC/B,OAAO,IAAI,MAAO,GAAE,IAAI,K;GADK;;;I;EAS/B,MAAM,KA  
AK,UAAU,SAAU,GAAE,qB;IAC/B,IAAI,QAAQ,SAAU,IAAG,E;IACzB,IAAI,KAAM,GAAE,CAAE,IAAG,EA  
AG,GAAE,KAAIB,C;MACE,MAAM,KAAK,CAAC,sBAAuB,GAAE,KAAIB,C;;IAGb,IAAI,IAAI,OAAO,EAAf,  
C;MACE,OAAO,G;;IAGT,IAAI,IAAI,WAAW,EAAhB,C;MACE,IAAI,IAAI,WAAW,CAAC,MAAM,KAAK,UA  
AZ,CAAnB,C;QAGE;;YAAI,YAA,Y,MAAM,KAAK,WAAW,CAAC,KAAD,C;QACtC,IAAI,MAAM,IAAI,IAAI,  
CAAC,SAAD,C;QACIB,IAAI,MAAM,GAAG,SAAS,CAAC,SAAD,CAAW,SAAS,CAAC,IAAD,C;QACIC,OAA  
O,GAAG,SAAS,CAAC,KAAD,CAAQ,GAAE,GAAG,MAAM,EAAE,SAAS,CAAC,KAAD,C;;QAEjD,OAAO,G  
AAI,GAAE,IAAI,OAAO,EAAE,SAAS,CAAC,KAAD,C;;;;;IAMvC,IAAI,eAAe,MAAM,KAAK,WAAW,CAAC,I  
AAI,IAAI,CAAC,KAAD,EAAQ,CAAR,CAAT,C;IAEzC,IAAI,MAAM,I;IACV,IAAI,SAAS,E;IACb,OAAO,IAAP  
,C;MACE,IAAI,SAAS,GAAG,IAAI,CAAC,YAAD,C;MACpB,IAAI,SAAS,GAAG,SAAS,CAAC,MAAM,SAAS,  
CAAC,YAAD,CAAhB,CAA+B,MAAM,E;MAC9D,IAAI,SAAS,MAAM,SAAS,CAAC,KAAD,C;MAE5B,GAAL,  
GAAE,M;MACN,IAAI,GAAG,OAAO,EAAd,C;QACE,OAAO,MAAO,GAAE,M;;QAEhB,OAAO,MAAM,OAAQ  
,GAAE,CAAvB,C;UACE,MAAO,GAAE,GAAL,GAAE,M;;QAEjB,MAAO,GAAE,EAAG,GAAE,MAAO,GAAE,  
M;;;GAzCE,0D;EAgD/B,MAAM,KAAK,UAAU,YAAa,GAAE,Y;IACIC,OAAO,IAAI,M;GADqB,yD;EAMIC,MA  
AM,KAAK,UAAU,WAA,Y,GAAE,Y;IACjC,OAAO,IAAI,K;GADoB,4D;EAMjC,MAAM,KAAK,UAAU,mBAAo  
B,GAAE,Y;IACzC,OAAQ,IAAI,KAAM,IAAG,CAAG,GACpB,IAAI,KADgB,GACR,MAAM,KAAK,gBAaIB,G  
AAE,IAAI,K;GAFX;;;I;EAUzC,MAAM,KAAK,UAAU,cAAe,GAAE,Y;IACpC,IAAI,IAAI,WAAW,EAAhB,C;M  
ACE,IAAI,IAAI,WAAW,CAAC,MAAM,KAAK,UAAZ,CAAnB,C;QACE,OAAO,E;;QAEp,OAAO,IAAI,OAAO,  
EAAE,cAAc,E;;MAGpC,IAAI,MAAM,IAAI,MAAO,IAAG,CAAE,GAAE,IAAI,MAAN,GAAe,IAAI,K;MAC7C,  
KAAK,IAAI,MAAM,EAAf,EAAMB,GAAL,GAAE,CAAzB,EAA4B,GAAG,EAA/B,C;QACE,IAAuB,CAAIB,GA  
AI,GAAG,CAAE,IAAG,GAAM,KAAG,CAAIB,C;UACE,K;;;MAGJ,OAAO,IAAI,MAAO,IAAG,CAAE,GAAE,  
GAAL,GAAE,EAAR,GAAa,GAAL,GAAE,C;;GAdV,mD;EAoBpC,MAAM,KAAK,UAAU,OAAQ,GAAE,Y;IAC7  
B,OAAO,IAAI,MAAO,IAAG,CAAE,IAAG,IAAI,KAAM,IAAG,C;GADZ,uD;EAM7B,MAAM,KAAK,UAAU,W  
AA,Y,GAAE,Y;IACjC,OAAO,IAAI,MAAO,GAAE,C;GADW,kD;EAMjC,MAAM,KAAK,UAAU,MAAO,GAAE,  
Y;IAC5B,OAAuB,CAAf,IAAI,KAAM,GAAE,CAAG,KAAG,C;GADA;;;I;EAS5B,MAAM,KAAK,UAAU,WAA,Y  
,GAAE,iB;IACjC,OAAQ,IAAI,MAAO,IAAG,KAAK,MAAQ,IAAI,IAAI,KAAM,IAAG,KAAK,K;GAD1B;;;I;EA  
SjC,MAAM,KAAK,UAAU,cAAe,GAAE,iB;IACpC,OAAQ,IAAI,MAAO,IAAG,KAAK,MAAQ,IAAI,IAAI,KAA  
M,IAAG,KAAK,K;GADvB;;;I;EASpC,MAAM,KAAK,UAAU,SAAU,GAAE,iB;IAC/B,OAAO,IAAI,QAAQ,CAA  
C,KAAD,CAAQ,GAAE,C;GADA;;;I;EAS/B,MAAM,KAAK,UAAU,gBAaIB,GAAE,iB;IACtC,OAAO,IAAI,QA  
AQ,CAAC,KAAD,CAAQ,IAAG,C;GADM;;;I;EAStC,MAAM,KAAK,UAAU,YAAa,GAAE,iB;IACIC,OAAO,IA  
AI,QAAQ,CAAC,KAAD,CAAQ,GAAE,C;GADG;;;I;EASIC,MAAM,KAAK,UAAU,mBAAoB,GAAE,iB;IACzC,  
OAAO,IAAI,QAAQ,CAAC,KAAD,CAAQ,IAAG,C;GADS;;;I;EAWzC,MAAM,KAAK,UAAU,QAAS,GAAE,iB  
;IAC9B,IAAI,IAAI,WAAW,CAAC,KAAD,CAAnB,C;MACE,OAAO,C;;IAGT,IAAI,UAAU,IAAI,WAAW,E;IAC  
7B,IAAI,WAAW,KAAK,WAAW,E;IAC/B,IAAI,OAAQ,IAAG,CAAC,QAAhB,C;MACE,OAAO,E;;IAET,IAAI,C  
AAC,OAAQ,IAAG,QAAhB,C;MACE,OAAO,C;;IAIT,IAAI,IAAI,SAAS,CAAC,KAAD,CAAO,WAAW,EAAhC,  
C;MACE,OAAO,E;;MAEP,OAAO,C;;GAIBmB,wD;EAwB9B,MAAM,KAAK,UAAU,OAAQ,GAAE,Y;IAC7B,I  
AAI,IAAI,WAAW,CAAC,MAAM,KAAK,UAAZ,CAAnB,C;MACE,OAAO,MAAM,KAAK,U;;MAEIB,OAAO,I  
AAI,IAAI,EAAE,IAAI,CAAC,MAAM,KAAK,IAAZ,C;;GAJI;;;I;EAc7B,MAAM,KAAK,UAAU,IAAK,GAAE,iB  
;IAG1B;QAAI,MAAM,IAAI,MAAO,KAAI,E;IACzB,IAAI,MAAM,IAAI,MAAO,GAAE,K;IACvB,IAAI,MAAM,  
IAAI,KAAM,KAAI,E;IACxB,IAAI,MAAM,IAAI,KAAM,GAAE,K;IAEtB,IAAI,MAAM,KAAK,MAAO,KAAI,E;  
IAC1B,IAAI,MAAM,KAAK,MAAO,GAAE,K;IACxB,IAAI,MAAM,KAAK,KAAM,KAAI,E;IACzB,IAAI,MAA  
M,KAAK,KAAM,GAAE,K;IAEvB,IAAI,MAAM,CAAV,EAAa,MAAM,CAAnB,EAASB,MAAM,CAA5B,EAA+  
B,MAAM,C;IACrC,GAAL,IAAG,GAAL,GAAE,G;IACb,GAAL,IAAG,GAAL,KAAI,E;IACf,GAAL,IAAG,K;IACP,  
GAAL,IAAG,GAAL,GAAE,G;IACb,GAAL,IAAG,GAAL,KAAI,E;IACf,GAAL,IAAG,K;IACP,GAAL,IAAG,GAAL,  
GAAE,G;IACb,GAAL,IAAG,GAAL,KAAI,E;IACf,GAAL,IAAG,K;IACP,GAAL,IAAG,GAAL,GAAE,G;IACb,GAA  
L,IAAG,K;IACP,OAAO,MAAM,KAAK,SAAS,CAAE,GAAL,IAAG,EAAL,GAAE,GAAL,EAAG,GAAL,IAAG,EA  
AI,GAAE,GAAL,C;GAzBH;;;I;EAK1B,MAAM,KAAK,UAAU,SAAU,GAAE,iB;IAC/B,OAAO,IAAI,IAAI,CA



,KAAK,KAAIB,EACI,IAAI,MAAO,GAAE,KAAK,MADtB,C;GADH;;;I;EAW1B,MAAM,KAAK,UAAU,GAAI,  
GAAE,iB;IACzB,OAAO,MAAM,KAAK,SAAS,CAAC,IAAI,KAAM,GAAE,KAAK,KAAIB,EACI,IAAI,MAAO,  
GAAE,KAAK,MADtB,C;GADJ;;;I;EAWzB,MAAM,KAAK,UAAU,IAAK,GAAE,iB;IAC1B,OAAO,MAAM,KA  
AK,SAAS,CAAC,IAAI,KAAM,GAAE,KAAK,KAAIB,EACI,IAAI,MAAO,GAAE,KAAK,MADtB,C;GADH;;;I;  
EAW1B,MAAM,KAAK,UAAU,UAAW,GAAE,mB;IAChC,OAAQ,IAAG,E;IACX,IAAI,OAAQ,IAAG,CAAf,C;  
MACE,OAAO,I;;MAEP,IAAI,MAAM,IAAI,K;MACd,IAAI,OAAQ,GAAE,EAAd,C;QACE,IAAI,OAAO,IAAI,M;  
QACf,OAAO,MAAM,KAAK,SAAS,CACvB,GAALI,IAAG,OADgB,EAETB,IAAK,IAAG,OAAS,GAAG,GAALI,KA  
AK,EAAG,GAAE,OAFZ,C;;QAI3B,OAAO,MAAM,KAAK,SAAS,CAAC,CAAD,EAALI,GAALI,IAAI,OAAQ,GAA  
E,EAAtB,C;;;GAZD;;;I;EAuBhC,MAAM,KAAK,UAAU,WAAY,GAAE,mB;IACjC,OAAQ,IAAG,E;IACX,IAAI,  
OAAQ,IAAG,CAAf,C;MACE,OAAO,I;;MAEP,IAAI,OAAO,IAAI,M;MACf,IAAI,OAAQ,GAAE,EAAd,C;QACE  
,IAAI,MAAM,IAAI,K;QACd,OAAO,MAAM,KAAK,SAAS,CACtB,GAALI,KAALI,OAAS,GAAG,IAAK,IAAI,EA  
AG,GAAE,OADZ,EAEvB,IAAK,IAAG,OAFc,C;;QAI3B,OAAO,MAAM,KAAK,SAAS,CACvB,IAAK,IAAI,OA  
AQ,GAAE,EADI,EAEvB,IAAK,IAAG,CAAE,GAAE,CAAF,GAAM,EAFO,C;;;GAZA;;;I;EA2BjC,MAAM,KA  
AK,UAAU,mBAaOB,GAAE,mB;IACzC,OAAQ,IAAG,E;IACX,IAAI,OAAQ,IAAG,CAAf,C;MACE,OAAO,I;;M  
AEP,IAAI,OAAO,IAAI,M;MACf,IAAI,OAAQ,GAAE,EAAd,C;QACE,IAAI,MAAM,IAAI,K;QACd,OAAO,MAA  
M,KAAK,SAAS,CACtB,GAALI,KAALI,OAAS,GAAG,IAAK,IAAI,EAAG,GAAE,OADZ,EAEvB,IAAK,KAALI,OA  
Fc,C;aAGtB,IAAI,OAAQ,IAAG,EAaf,C;QACL,OAAO,MAAM,KAAK,SAAS,CAAC,IAAD,EAAO,CAAP,C;;Q  
AE3B,OAAO,MAAM,KAAK,SAAS,CAAC,IAAK,KAAK,OAAQ,GAAE,EAAtB,EAA0B,CAA1B,C;;;GAdQ;A,E  
AoBzC,MAAM,KAAK,UAAU,OAAQ,GAAE,iB;IAC3B,OAAO,KAAM,YAAW,MAAM,KAAM,IAAG,IAAI,W  
AAW,CAAC,KAAD,C;G;EAG1D,MAAM,KAAK,UAAU,gBAaiB,GAAE,MAAM,KAAK,UAAU,Q;EAE7D,MA  
AM,KAAK,UAAU,IAAK,GAAE,Y;IACxB,OAAO,IAAI,IAAI,CAAC,MAAM,KAAK,IAAZ,C;G;EAGnB,MAA  
M,KAAK,UAAU,IAAK,GAAE,Y;IACxB,OAAO,IAAI,IAAI,CAAC,MAAM,KAAK,QAAs,C;G;EAGnB,MAAM,  
KAAK,UAAU,QAAS,GAAE,Y;IAC5B,OAAO,IAAI,SAAS,E;G;EAGxB,MAAM,KAAK,UAAU,UAAW,GAAE,  
Y;IAC9B,OAAO,I;G;EAGX,MAAM,KAAK,UAAU,WAAY,GAAE,MAAM,KAAK,UAAU,O;EACxD,MAAM,K  
AAK,UAAU,IAAK,GAAE,MAAM,KAAK,UAAU,I;EAEjD,MAAM,KAAK,UAAU,QAAS,GAAE,iB;IAC5B,OA  
AO,IAAI,MAAM,OAAO,OAAO,UAAxB,CAAmC,IAAnC,EAAYC,KAAzC,C;G;EC1zBS;;;IAApB,MAAM,aA  
Ac,GAAE,2B;G;EAGtB,MAAM,qBAAsB,GAAE,oB;IAC1B,OAAO,G;G;EAGX,MAAM,aAAc,GAAE,e;IACIB,I  
AAI,IAAI,Y;MACJ,CAAE,GAAE,GAAG,E;MACP,OAAO,CAAC,MAAM,CAAC,IAAD,EAAO,SAAP,C;K;IAEI  
B,OAAO,Y;MACH,OAAO,CAAC,MAAM,CAAC,IAAD,EAAO,SAAP,C;K;G;EAItB,MAAM,SAAU,GAAE,gB;I  
ACd,OAAO,kB;MACH,OAAO,OAAO,MAAO,KAAI,I;K;G;EAIjC,MAAM,aAAc,GAAE,iB;IACIB,OAAO,kB;M  
ACH,OAAO,MAAM,OAAO,CAAC,MAAD,EAAS,KAAT,C;K;G;EAI5B,MAAM,OAAQ,GAAE,c;IACZ,OAAO,  
kB;MACH,OAAO,MAAO,IAAG,IAAK,IAAG,EAAE,CAAC,MAAD,C;K;G;EAIInC,MAAM,aAAc,GAAE,gB;IA  
CIB,OAAO,kB;MACH,OAAO,CAAC,CAAC,MAAD,CAAS,IAAG,CAAC,CAAC,MAAD,C;K;G;EAI7B,MAAM,  
qBAAsB,GAAE,wC;G;EAG9B,MAAM,YAAa,GAAE,iB;IACjB,OAAO,K;G;EAGX,MAAM,gBAaiB,GAAE,qB;  
IACrB,gBAAgB,E;G;EAGpB,MAAM,oBAaQb,GAAE,qB;IACzB,gBAAgB,E;G;EAGpB,MAAM,kBAAmB,GAA  
E,qB;IACvB,gBAAgB,E;G;EAGpB,MAAM,mBAaOB,GAAE,4B;IACxB,gBAAgB,E;G;EAGpB,MAAM,6BAA8  
B,GAAE,yB;IACIC,gBAAgB,E;G;EAGpB,4B;IACI,MAAM,IAAI,KAAJ,CACF,iDAakD,GACID,qDAAsD,GACt  
D,uDAHE,C;G;EAMV,MAAM,gBAaiB,GAAE,4B;IACrB,OAAO,Y;MACH,OAAO,Y;K;G;ECjFE;;;IAAjB,MAA  
M,UAAW,GAAE,gB;IACf,IAAI,QAAQ,OAAO,C;IACnB,IAAI,KAAM,KAAI,QAAd,C;MACI,IAAI,OAAO,CAA  
E,KAAI,QAajB,C;QACI,OAAO,MAAM,gBAAgB,CAAC,CAAD,EAALI,CAAJ,C;;MAEjC,OAAO,MAAM,mBA  
AmB,CAAC,CAAD,EAALI,CAAJ,C;;IAEpC,IAAI,KAAM,KAAI,QAAS,IAAG,KAAM,KAAI,SAApC,C;MACI,O  
AAO,MAAM,mBAAmB,CAAC,CAAD,EAALI,CAAJ,C;;IAEpC,OAAO,CAAC,gBAAgB,CAAC,CAAD,C;G;EAG  
5B,MAAM,mBAaOB,GAAE,gB;IACxB,OAAO,CAAE,GAAE,CAAE,GAAE,EAaf,GAAO,CAAE,GAAE,CAAE  
,GAAE,CAAF,GAAM,C;G;EAGpC,MAAM,gBAaiB,GAAE,gB;IACrB,IAAI,CAAE,GAAE,CAAR,C;MAAW,O  
AAO,E;IACIB,IAAI,CAAE,GAAE,CAAR,C;MAAW,OAAO,C;IAEIB,IAAI,CAAE,KAAI,CAAV,C;MACI,IAAI,  
CAAE,KAAI,CAAV,C;QAAa,OAAO,C;MAEpB,IAAI,KAAK,CAAE,GAAE,C;MACb,OAAO,EAAG,KAAI,CA  
AE,GAAE,CAAE,GAAE,CAAF,GAAG,EAAG,GAAE,CAAE,GAAE,EAaf,GAAO,C;;IAG7C,OAAO,CAAE,KA  
AI,CAAE,GAAG,CAAE,KAAI,CAAE,GAAE,CAAF,GAAM,CAAjB,GAAsB,E;G;EAGzC,MAAM,QAAS,GAAE

,iB;IACb,OAAO,MAAM,OAAO,CAAC,KAAK,GAAC,CAAP,C;G;EAGxB,MAAM,QAAS,GAAE,iB;IACb,OAAO,MAAM,OAAO,CAAC,KAAK,GAAC,CAAP,C;G;EAGxB,MAAM,KAAM,GAAE,IAAI,KAAM,IAAG,I;EAE3B,MAAM,aAAc,GAAE,I;EAEtB,oB;IACI,OAAyB,CAAhB,CAAE,GAAE,YAAY,KAAG,CAAE,GAAE,KAAP,CAAE,GAAE,CAAZ,CAAE,GAAE,KAAQ,KAAG,CAAE,GAAE,CAAP,CAAW,GAAE,C;G;EA6DtE,CA1DD,Y;IACG,IAAI,MAAM,IAAI,WAAJ,CAAgB,CAAhB,C;IACV,IAAI,aAAa,IAAI,YAAJ,CAAI,CAAjB,C;IACjB,IAAI,aAAa,IAAI,YAAJ,CAAI,CAAjB,C;IACjB,IAAI,WAAW,IAAI,UAAJ,CAAE,GAAf,C;IACf,IAAI,WAAW,C;IACf,IAAI,YAAY,C;IAEHb,UAAU,CAAC,CAAD,CAAI,GAAE,EAAf,A,IACd,IAAI,QAAQ,CAAC,QAAD,CAAW,KAAl,CAA3B,C;MACI,QAAS,GAAE,C;MACX,SAAU,GAAE,C;;IAGhB,MAAM,aAAc,GAAE,iB;MACIB,OAAO,MAAM,gBAAgB,CAAC,KAAK,CAAC,KAAD,CAAQ,GAAE,GAAf,GAAQ,KAAtB,C;K;IAGjC,MAAM,gBAAiB,GAAE,iB;MACrB,UAAU,CAAC,CAAD,CAAI,GAAE,K;MACHb,OAAO,MAAM,KAAK,SAAS,CAAC,QA AQ,CAAC,QAAD,CAAT,EAAqB,QAAQ,CAAC,SAAD,CAA7B,C;K;IAG/B,MAAM,eAAgB,GAAE,iB;MACpB,QAAQ,CAAC,QAAD,CAAW,GAAE,KAAK,K;MAC1B,QAAQ,CAAC,SAAD,CAAY,GAAE,KAAK,M;MAC3B,OAAO,UAAU,CAAC,CAAD,C;K;IAGrB,MAAM,YAAa,GAAE,iB;MACjB,OAAO,MAAM,eAAe,CAAC,KAAK,CAAC,KAAD,CAAQ,GAAE,GAAf,GAAQ,KAAtB,C;K;IAGhC,MAAM,eAAgB,GAAE,iB;MACpB,UAAU,CAAC,CAAD,CAAI,GAAE,K;MACHb,OAAO,QAAQ,CAAC,CAAD,C;K;IAGnB,MAAM,cAAe,GAAE,iB;MACnB,QAAQ,CAAC,CAAD,CAAI,GAAE,K;MACd,OAAO,UAAU,CAAC,CAAD,C;KAFa,A,IAMrB,MAAM,cAAe,GAAE,iB;MACnB,UAAU,CAAC,CAAD,CAAI,GAAE,K;MACHb,OAAO,QAAQ,CAAC,SAAD,CAAY,GAAE,a;K;IAGjC,MAAM,eAAgB,GAAE,e;MACpB,IAAc,CAAT,GAAl,GAAE,CAAG,MAAI,GAAlB,C;QACI,OAAO,GAAl,GAAE,C;;QAGb,UAAU,CAAC,CAAD,CAAI,GAAE,G;QACHb,OAAc,CAA9B,QAAQ,CAAC,SAAD,CAAY,GAAE,EAAg,GAAE,CAAG,IAAE,QAAQ,CAAC,QAAD,CAAW,GAAE,C;;K;GAGvE,G;EAEF,MAAM,cAAe,GAAE,a;IACnB,OAAO,CAAE,IAAG,IAAK,GAAE,CAAF,GAAM,MAAM,SAAS,E;G;EC7G1C;;QAAI,OAAO,MAAM,UAAU,WAAy,KAAl,WAA3C,C;IACI,MAAM,eAAe,CAAC,MAAM,UAAp,EAAmB,YAAnB,EAAiC,CACID,KADkD,EAC3C,kC;MACH,QAAS,GAAE,QAAS,IAAG,C;MACvB,OAAO,IAAI,YAAY,CAAC,YAAD,EAAe,QAAf,CAAyB,KAAl,Q;KAHN,CAAjC,C;;EAOzB,IAAI,OAAO,MAAM,UAAU,SAAU,KAAl,WAAzC,C;IACI,MAAM,eAAe,CAAC,MAAM,UAAp,EAAmB,UAAAnB,EAA+B,CACHd,KADgD,EACzC,kC;MACH,IAAI,gBAAgB,IAAI,SAAS,E;MACjC,IAAI,QAAS,KAAl,SAAU,IAAG,QAAS,GAAE,aAAa,OAAtD,C;QACI,QAAS,GAAE,aAAa,O;;MAE5B,QAAS,IAAG,YAAY,O;MACxB,IAAI,YAAY,aAAa,QAAQ,CAAC,YAAD,EAAe,QAAf,C;MACrC,OAAO,SAAU,KAAl,EAAg,IAAG,SAAU,KAAl,Q;KARG,CAA/B,C;;EAAzB,IAAI,OAAO,IAAI,KAAm,KAAl,WAAzB,C;IACI,IAAI,KAAM,GAAE,a;MACR,CAAE,GAAE,CAAC,CAAH;A,MACF,IAAI,CAAE,KAAI,CAAE,IAAG,KAAK,CAAC,CAAD,CAApB,C;QACI,OAAO,MAAM,CAAC,CAAD,C;;MAEjB,OAAO,CAAE,GAAE,CAAE,GAAE,CAAF,GAAM,E;K;;EAG3B,IAAI,OAAO,IAAI,MAAO,KAAl,WAA1B,C;IACI,IAAI,MAAO,GAAE,a;MACT,IAAI,KAAK,CAAC,CAAD,CAAT,C;QACI,OAAO,G;;MAEX,IAAI,CAAE,GAAE,CAAR,C;QACI,OAAO,IAAI,MAAM,CAAC,CAAD,C;;MAErB,OAAO,IAAI,KAAK,CAAC,CAAD,C;K;;EAKtB,CAnKD,Y;IACG,IAAI,UAAU,qB;IACd,IAAI,iBAAiB,IAAI,KAAK,CAAC,OAAD,C;IAC9B,IAAI,iBAAiB,IAAI,KAAK,CAAC,cAAD,C;IAC9B,IAAI,uBAAuB,CAAC,GAAC,c;IAC7B,IAAI,uBAAuB,CAAC,GAAC,c;IAE7B,IAAI,OA AO,IAAI,KAAM,KAAl,WAAzB,C;MACI,IAAI,KAAM,GAAE,a;QACR,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAIB,C;UACI,IAAI,SAAS,C;UACb,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAIB,C;YACI,MAAO,IAAI,CAAE,GAAE,CAAE,GAAE,CAAG,GAAE,C;;UAE5B,OAAO,M;;UAEP,IAAI,IAAI,IAAI,IAAI,CAAC,CAAD,C;UACHb,IAAI,KAAK,CAAE,GAAE,C;UACb,IAAI,CAAC,QAAQ,CAAC,CAAD,CAAb,C;YAAkB,OAAO,IAAI,IAAI,CAAC,CAAE,GAAE,IAAI,IAAT,C;UACjC,IAAI,CAAC,QAAQ,CAAC,EAAD,CAAb,C;YAAmB,OA AO,CAAC,IAAI,IAAI,CAAC,CAAC,CAAE,GAAE,IAAI,IAAV,C;UACnC,OAAGB,CAAR,CAAE,GAAE,EAAl,IAAE,C;;O;;IAI9B,IAAI,OAAO,IAAI,KAAM,KAAl,WAAzB,C;MACI,IAAI,KAAM,GAAE,a;QACR,IAAI,IAAI,IAAI,IAAI,CAAC,CAAD,C;QACHb,IAAI,KAAK,CAAE,GAAE,C;QACb,IAAI,CAAC,QAAQ,CAAC,CAAD,CAAl,IAAG,CAAC,QAAQ,CAAC,EAAD,CAA7B,C;UAAmC,OAAO,IAAI,IAAI,CAAC,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,IAAI,IAAnB,C;QACID,OAAGB,CAAR,CAAE,GAAE,EAAl,IAAE,C;O;;IAI1B,IAAI,OAAO,IAAI,KAAM,KAAl,WAAzB,C;MACI,IAAI,KAAM,GAAE,a;QACR,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAIB,C;UACI,IAAI,SAAS,C;UACb,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAIB,C;YACI,MAAO,IAAI,CAAE,GAAE,CAAE,GAAE,CAAG,GAAE,C;;UAE5B,OAAO,M;;UAGP,IAAI,IAAI,IAAI,IAAI,CAAC,CAAC,CA

AF,CAAhB,EAAsB,IAAI,IAAI,IAAI,CAAC,CAAC,CAAF,C;UACIC,OAAO,CAAE,KAAI,QAAS,GAAE,CAAF,GAAM,CAAE,KAAI,QAAS,GAAE,EAFF,GAAe,CAAP,CAAE,GAAE,CAAG,KAAG,CAAE,GAAE,CAAP,C;;O;;;IAQtE,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;MACI,IAAI,QAAQ,a;QACR,IAAI,CAAE,IAAG,CAAC,cAAV,C;UAEL,IAAI,CAAE,GAAE,oBAAR,C;YAEI,IAAI,CAAE,GAAE,oBAAR,C;cAGI;qBAAO,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,IAAI,I;;cAKzB;qBAAO,IAAI,IAAI,CAAC,CAAE,GAAE,CAAE,GAAG,CAAE,IAAG,CAAE,GAAE,CAAP,CAAZ,C;;YAKnB,OAAO,IAAI,IAAI,CAAC,CAAE,GAAE,IAAI,KAAK,CAAC,CAAE,GAAE,CAAE,GAAE,CAAT,CAAd,C;;eAGIB,IAAI,CAAE,IAAG,CAAC,cAAV,C;UAED,OAAO,CAAC,KAAK,CAAC,CAAC,CAAF,C;;UAKb;cAAI,SAAS,C;UACb,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,IAAG,cAAAnB,C;YAEI,IAAI,KAAK,CAAE,GAAE,CAAE,GAAE,CAAjB;A,YAEA,MAAO,IAAG,EAAG,GAAE,C;;UAEnB,OAAO,M;;O;MAGf,IAAI,MAAO,GAAE,K;;IAEjB,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;MACI,IAAI,MAAO,GAAE,a;QACT,IAAI,CAAE,GAAE,CAAR,C;UAEL,OAAO,G;eAEN,IAAI,CAAE,GAAE,CAAE,IAAG,cAAb,C;UAED,IAAI,CAAE,GAAE,oBAAR,C;YAGI;mBAAO,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,IAAI,I;;YAIzB,OAAO,IAAI,IAAI,CAAC,CAAE,GAAE,IAAI,KAAK,CAAC,CAAE,GAAE,CAAE,GAAE,CAAT,CAAd,C;;UAKnB,IAAI,IAAI,IAAI,KAAK,CAAC,CAAE,GAAE,CAAL,CAAjB;A,UAEA,IAAI,SAAS,C;UACb,IAAI,CAAE,IAAG,cAAT,C;YAEI,IAAI,KAAK,CAAE,GAAE,CAAE,GAAE,CAAjB;A,YAEA,MAAO,IAAG,EAAG,GAAE,E;;UAGnB,OAAO,IAAI,KAAK,CAAC,CAAD,CAAI,GAAE,M;;O;;IAIIC,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;MACI,IAAI,MAAO,GAAE,a;QACT,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAIB,C;UACI,IAAI,SAAS,C;UACb,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAIB,C;YACI,MAAO,IAAI,CAAE,GAAE,CAAE,GAAE,CAAG,GAAE,C;;UAE5B,OAAO,M;;QAEX,OAAO,IAAI,IAAI,CAAS,CAAP,CAAE,GAAE,CAAG,KAAG,CAAE,GAAE,CAAP,CAAT,CAAoB,GAAE,C;O;;IAG7C,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;MACI,IAAI,MAAO,GAAE,a;QACT,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAIB,C;UACI,IAAI,KAAK,CAAE,GAAE,C;UACb,IAAI,KAAK,EAAG,GAAE,C;UACd,IAAI,KAAK,EAAG,GAAE,CAAd;A,UAEA,OAAQ,CAAC,EAAG,GAAE,CAAE,GAAE,EAAG,GAAE,CAAE,GAAE,EAAG,GAAE,CAAE,GAAE,C;;QAExC,OAAO,IAAI,IAAI,CAAC,CAAE,GAAE,CAAL,C;O;;IAGvB,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;MACI,IAAI,MAAO,GAAE,a;QACT,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAIB,C;UACI,IAAI,KAAK,CAAE,GAAE,C;UACb,IAAI,KAAK,EAAG,GAAE,C;UACd,IAAI,KAAK,EAAG,GAAE,CAAd;A,UAEA,OAAQ,EAAG,GAAE,EAAG,GAAE,EAAG,GAAE,CAAE,GAAE,EAAG,GAAE,CAAE,GAAE,C;;QAExC,OAAO,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,C;O;;GAG/B,G;EACF,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;IACI,IAAI,MAAO,GAAE,Y;MACT,IAAI,IAAI,C;MACR,IAAI,SAAS,SAAS,O;MAEtB,KAAK,IAAI,IAAI,CAAAb,EAAGB,CAAE,GAAE,MAApB,EAA4B,CAAC,EAA7B,C;QACI,IAAI,SAAS,CAAC,CAAD,CAAI,KAAI,QAAS,IAAG,SAAS,CAAC,CAAD,CAAI,KAAI,CAAC,QAAnD,C;UACI,OAAO,Q;;QAEX,CAAE,IAAG,SAAS,CAAC,CAAD,CAAI,GAAE,SAAS,CAAC,CAAD,C;;MAEjC,OAAO,IAAI,KAAK,CAAC,CAAD,C;K;;EAGxB,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;IACI,IAAI,MAAO,GAAE,a;MACT,OAAO,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,IAAI,O;K;;EAGjC,IAAI,OAAO,IAAI,KAAK,IAAI,MAAO,KAAI,WAA1B,C;IACI,IAAI,MAAO,GAAG,oB;MACV,OAAO,a;QACH,IAAI,SAAS,CAAE,KAAI,C;QACnB,IAAI,MAAO,KAAI,CAAF,C;UACI,OAAO,E;;QAEX,OAAO,EAAG,IAAG,GAAG,CAAC,MAAD,CAAS,GAAE,GAAL,GAAE,CAAvB,CAA0B,GAAE,CAAtC;A,O;KAEN,CAAC,IAAI,IAAL,EAAW,IAAI,IAAf,C;;EAIN,IAAI,OAAO,WAAW,OAAQ,KAAI,WAAIC,C;IACI,WAAW,OAAQ,GAAE,a;MACjB,OAAO,CAAE,IAAG,IAAK,IAAG,CAAC,UAAW,IAAG,IAAK,IAAG,CAAC,UAAU,UAAW,KAAI,SAAS,UAAU,U;K;;EAIhG,IAAI,OAAO,KAAK,UAAU,KAAM,KAAI,WAApC,C;IAEyB;IAArB,MAAM,eAAe,C AAC,KAAK,UAAU,EAakB,MAAIB,EAA0B,CAC3C,KAD2C,EACpC,iB;MAGH;UAAI,IAAK,IAAG,IAAZ,C;QACI,MAAM,IAAI,SAAJ,CAAc,6BAAd,C;;MAGV,IAAI,IAAI,MAAM,CAAC,IAAD,CAAd;A,MAGA,IAAI,MAAM,CAAC,OAAQ,KAAI,CAAvB;A,MAGA,IAAI,QAAQ,SAAS,CAAC,CAAD,C;MACrB,IAAI,gBAAGB,KAAI,IAAG,CAA7B;A,MAGA,IAAI,IAAI,aAAc.GAAE,CAAE,GACIB,IAAI,IAAI,CAAC,GAAL,GAAE,aAAP,EAA sB,CAAtB,CADU,GAEIB,IAAI,IAAI,CAAC,aAAD,EAAGB,GAAhB,CAFhB;A,MAKA,IAAI,MAAM,SAAS,CAAC,CAAD,C;MACnB,IAAI,cAAc.GAAI,KAAI,SAAU,GACIB,GADkB,GACZ,GAAL,IAAG,CAD/B;A,MAIA,IAAI,aAAa,WAAW,GAAE,CAAE,GACHB,IAAI,IAAI,CAAC,GAAL,GAAE,WAAW,EAAB,CAApB,CADQ,GAeHb,IAAI,IAAI,CAAC,WAAD,EAAC,GAAd,CAFzB;A,MAKA,OAAO,CAAE,GAAE,UAAW,C;QACI,CAAC,CAAC,



CAAD,CAAI,GAAE,K;QACP,CAAC,E;;;MAIL,OAAO,C;KAvCgC,CAA1B,C;;EA4HvB,CAhFD,Y;IACG,yC;M  
ACI,IAAI,MAAO,GAAE,CAAb,C;QAAGB,OAAO,IAAI,IAAI,CAAC,CAAD,EAAI,MAAO,GAAE,MAAb,C;MA  
C/B,OAAO,IAAI,IAAI,CAAC,MAAD,EAAS,MAAT,C;K;IAEnB,qC;MACI,IAAI,OAAO,GAAl,KAAl,WAAAb,  
C;QACI,GAAl,GAAE,IAAI,O;;MAEd,KAAM,GAAE,eAAe,CAAC,KAAM,IAAG,CAAV,EAAa,IAAI,OAAjB,C;  
MACvB,GAAl,GAAE,IAAI,IAAI,CAAC,KAAD,EAAQ,eAAe,CAAC,GAAD,EAAM,IAAI,OAAV,CAAvB,C;M  
ACd,OAAO,IAAI,IAAI,YAAR,CAAqB,IAAI,SAAS,CAAC,KAAD,EAAQ,GAAR,CAAI,C,C;K;IAGX,IAAI,SA  
S,CAAC,SAAD,EAAY,UAAZ,EAAwB,WAAxB,EAAqC,UAArC,EAAiD,YAAjD,EAA+D,YAA/D,C;IACb,KAA  
K,IAAI,IAAI,CAAb,EAAGB,CAAE,GAAE,MAAM,OAA1B,EAAMC,EAAE,CAArC,C;MACI,IAAI,aAAa,MAA  
M,CAAC,CAAD,C;MACvB,IAAI,OAAO,UAAU,UAAU,KAAM,KAAl,WAAzC,C;QACI,MAAM,eAAe,CAAC,  
UAAU,UAAx,EAAuB,MAAvB,EAA+B,CAChD,KADgD,EACzC,KAAK,UAAU,KAD0B,CAA/B,C;;MAIzB,IA  
AI,OAAO,UAAU,UAAU,MAAO,KAAl,WAA1C,C;QACI,MAAM,eAAe,CAAC,UAAU,UAAx,EAAuB,OAAvB,  
EAAGC,CACjD,KADiD,EAC1C,eAD0C,CAAhC,C;;;MAQJ,CAApB,Y;OAAc,MAAM,CAAC,IAAD,EAAO,IA  
AI,UAAJ,CAAe,CAAf,CAAP,E;;MAErB,IAAI,QAAQ,QAAQ,UAAU,M;MAC9B,MAAM,eAAe,CAAC,QAAQ,U  
AAT,EAAqB,OAArB,EAA8B,CAC/C,KAD+C,EACxC,uB;QACH,OAAO,KAAK,KAAK,CAAC,IAAD,EAAO,IA  
AP,EAAa,EAAE,MAAM,KAAK,CAAC,KAAD,CAA1B,C;OAF0B,CAA9B,C;;IASzB,KAAK,IAAI,IAAI,CAAb,  
EAAGB,CAAE,GAAE,MAAM,OAA1B,EAAMC,EAAE,CAArC,C;MACI,IAAI,aAAa,MAAM,CAAC,CAAD,C;M  
ACvB,IAAI,OAAO,UAAU,UAAU,IAAK,KAAl,WAAxC,C;QACI,MAAM,eAAe,CAAC,UAAU,UAAx,EAAuB,  
KAAvB,EAA8B,CAC/C,KAD+C,EACxC,0B;UACH,OAAO,EAAE,MAAM,KAAK,CAAC,IAAD,CAAM,IAAI,C  
AAC,QAAD,EAAW,IAAX,C;SAFa,CAA9B,C;;;IAU7B,IAAI,uBAAuB,gB;MACvB,IAAI,CAAE,GAAE,CAAR,  
C;QAAW,OAAO,E;MACIB,IAAI,CAAE,GAAE,CAAR,C;QAAW,OAAO,C;MAEIB,IAAI,CAAE,KAAl,CAAV,  
C;QACI,IAAI,CAAE,KAAl,CAAV,C;UAAa,OAAO,C;QAEpB,IAAI,KAAK,CAAE,GAAE,C;QACb,OAAO,EA  
G,KAAl,CAAE,GAAE,CAAE,GAAE,CAAF,GAAG,EAAG,GAAE,CAAE,GAAE,EAAG,GAAG,C;;MAG7C,OA  
AO,CAAE,KAAl,CAAE,GAAG,CAAE,KAAl,CAAE,GAAE,CAAF,GAAM,CAAjB,GAAsB,E;K;IAGzC,KAAK,I  
AAI,IAAI,CAAb,EAAGB,CAAE,GAAE,MAAM,OAA1B,EAAMC,EAAE,CAArC,C;MACI,IAAI,aAAa,MAAM,C  
AAC,CAAD,C;MACvB,IAAI,OAAO,UAAU,UAAU,KAAM,KAAl,WAAzC,C;QACI,MAAM,eAAe,CAAC,UAA  
U,UAAx,EAAuB,MAAvB,EAA+B,CAChD,KADgD,EACzC,2B;UACH,OAAO,KAAK,UAAU,KAAK,KAAK,C  
AAC,IAAD,EAAO,eAAgB,IAAG,oBAA1B,C;SAFY,CAA/B,C;;GAO/B,G;ECXU;;;IAAZ,MAAM,KAAM,GAA  
E,CACV,KADU,EACH,OADG,EAEV,SAFU,EAEC,WAFD,EAGV,MAHU,EAGF,QAHE,C;EAMd,MAAM,WA  
AY,GAAE,2C;IACHB,IAAI,qBAAqB,MAAM,yBAAyB,CAAC,KAAD,EAAQ,YAAR,C;IACxD,IAAI,kBAAmB,I  
AAG,IAAK,IAAG,kBAAkB,IAAK,IAAG,IAA5D,C;MACI,OAAO,kBAAkB,IAAI,KAAK,CAAC,UAAD,C;;IAGt  
C,kBAAmB,GAAE,MAAM,yBAAyB,CAAC,UAAD,EAAa,YAAb,C;IACpD,IAAI,kBAAmB,IAAG,IAAK,IAAG,  
OAAQ,IAAG,kBAA7C,C;MACI,OAAO,UAAU,CAAC,YAAD,C;;IAGrB,OAAO,MAAM,WAAW,CAAC,UAAD,  
EAAa,MAAM,eAAe,CAAC,KAAD,CAAI,C,EAA2C,YAA3C,C;G;EAG5B,MAAM,WAAW,GAAE,kD;IACHB,IA  
AI,qBAAqB,MAAM,yBAAyB,CAAC,KAAD,EAAQ,YAAR,C;IACxD,IAAI,kBAAmB,IAAG,IAAK,IAAG,kBAA  
kB,IAAK,IAAG,IAA5D,C;MACI,kBAAkB,IAAI,KAAK,CAAC,UAAD,EAAa,KAAb,C;MAC3B,M;;IAGj,kBAA  
mB,GAAE,MAAM,yBAAyB,CAAC,UAAD,EAAa,YAAb,C;IACpD,IAAI,kBAAmB,IAAG,IAAK,IAAG,OAAQ,I  
AAG,kBAA7C,C;MACI,UAAU,CAAC,YAAD,CAAe,GAAE,K;MAC3B,M;;IAGJ,MAAM,WAAW,CAAC,UAA  
D,EAAa,MAAM,eAAe,CAAC,KAAD,CAAI,C,EAA2C,YAA3C,EAAYD,KAAzD,C;G;EAGrB,iD;IACI,IAAI,IAA  
K,KAAl,KAAb,C;MAAoB,OAAO,I;IAE3B,IAAI,WAAW,IAAI,W;IACnB,IAAI,QAAS,IAAG,IAAhB,C;MACI,I  
AAI,aAAa,QAAQ,W;MACzB,KAAK,IAAI,IAAI,CAAb,EAAGB,CAAE,GAAE,UAAU,OAA9B,EAAuC,CAAC,E  
AAxC,C;QACI,IAAI,0BAA0B,CAAC,UAAU,CAAC,CAAD,CAAX,EAAGB,KAAhB,CAA9B,C;UACI,OAAO,I;;  
;IAKnB,IAAI,iBAAiB,IAAI,UAAW,IAAG,IAAK,GAAE,MAAM,eAAe,CAAC,IAAI,UAAAL,CAAvB,GAA0C,I;I  
ACtF,IAAI,mBAAmB,cAAe,IAAG,IAAK,GAAE,cAAc,YAAhB,GAA+B,I;IAC7E,OAAO,gBAAiB,IAAG,IAAK,  
IAAG,0BAA0B,CAAC,gBAAD,EAAMB,KAAAb,C;G;EASnD;;;IAAd,MAAM,OAAQ,GAAE,yB;IACZ,IAAI,K  
AAM,KAAl,MAAd,C;MACI,QAAQ,OAAO,MAAf,C;aACS,Q;aACA,Q;aACA,S;aACA,U;UACD,OAAO,I;;UAE  
P,OAAO,MAAO,YAAW,M;;IAIrC,IAAI,MAAO,IAAG,IAAK,IAAG,KAAM,IAAG,IAAK,KAAl,OAAO,MAAO  
,KAAl,QAAS,IAAG,OAAO,MAAO,KAAl,UAApD,CAApC,C;MACI,OAAO,K;;IAGX,IAAI,OAAO,KAAM,KA  
Al,UAAW,IAAG,MAAO,YAAW,KAArD,C;MACI,OAAO,I;;IAGX,IAAI,QAAQ,MAAM,eAAe,CAAC,KAAD,C;

IACjC,IAAI,cAAc,KAAM,IAAG,IAAK,GAAE,KAAK,YAAP,GAAzB,I;IACtD,IAAI,WAAY,IAAG,IAAK,IAAG  
,YAAa,IAAG,WAA3C,C;MACI,IAAI,WAAY,WAAY,W;MAC1B,IAAI,QAAQ,KAAM,KAAL,MAAM,KAAK,O  
AAjC,C;QACI,OAAO,MAAO,KAAL,K;;IAI1B,IAAI,gBAAgB,KAAK,WAAzB;A,IAGA,IAAI,aAAc,IAAG,IAAr  
B,C;MACI,OAAO,MAAO,YAAW,K;;IAG7B,IAAI,aAAa,KAAM,KAAL,MAAM,KAAK,UAAW,IAAG,MAAM,  
YAAa,IAAG,IAA1E,C;MACI,OAAO,0BAA0B,CAAC,MAAM,YAAP,EAAqB,KAArB,C;;IAGrC,OAAO,K;G;EA  
GX,MAAM,SAAU,GAAE,a;IACd,OAAO,OAAO,CAAE,IAAG,QAAS,IAAG,CAAE,YAAW,MAAM,K;G;EAGt  
D,MAAM,OAAQ,GAAE,iB;IACZ,OAAO,KAAM,YAAW,MAAM,U;G;EAGIC,MAAM,aAAc,GAAE,iB;IACIB,I  
AAI,OAAO,OAAO,K;IAEIB,OAAO,IAAK,KAAL,QAAS,IACIB,IAAK,KAAL,SAAU,IACnB,MAAM,SAAS,CAA  
C,KAAD,CAAQ,IACvB,MAAM,OAAO,CAAC,KAAD,EAAQ,MAAM,OAAO,WAArB,C;G;EAGxB,MAAM,eA  
AgB,GAAE,iB;IACpB,OAAO,OAAO,KAAM,KAAL,QAAS,IAAG,MAAM,OAAO,CAAC,KAAD,EAAQ,MAAM  
,OAAO,aAArB,C;G;;aCnDV,gB;;;ICrE3C,gB;MAkBI,4B;MAjBA,aAA6C,E;MAC7C,gBAAGD,C;K;4EAG5  
C,Y;MAAQ,iB;K;+EAGR,Y;MAAQ,oB;K;qCAEZ,iB;MAAyC,OAAQ,0BAAR,YAAQ,EAAU,KAAM,QAAbB,C;  
K;4BAEjD,iB;MAAmC,gBAAS,K;K;8BAE5C,Y;MAA+B,OAAnc,MAAmC,kBAA8B,IAA9B,C;K;8BAE/B,Y;M  
AA0B,gB;K;IAE1B,0B;MAAA,8B;K;;IAAA,sC;MAAA,qC;QAAA,oB;;MAAA,8B;K;;IDfJ,mC;MAC4C,oBAAa,  
MAAS,IAAT,CAAb,EAA6B,SAA7B,C;K;gEAE5C,yB;MAAA,mB;MAAA,6B;QAC2D,YAAa,QAAS,IAAT,C;Q  
AIvD,Q;QAAA,OAAA,KAAM,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,iB;UACI,MAAM,CAAN,IALgF,IAK  
rE,CAAK,CAAL,C;;QALwC,OAOhD,K;O;KARX,C;gEAGA,uB;MAEiB,Q;MAAA,OAAA,KAAM,OAAN,GAAa  
,CAAb,I;MAAb,aAAU,CAAV,iB;QACI,MAAM,CAAN,IAAW,KAAK,CAAL,C;;MAEf,OAAO,K;K;IAGX,kC;M  
AIiB,IAAN,I;MAFP,aAAsB,MAAe,IAAf,C;MACtB,gBAAkB,c;MAEd,IADS,IACT,mBADs,IACT,EAAM,IAAN,  
E;QAAC,oBAAa,MAAb,EAAqB,KAArB,C;WACd,WAFS,IAET,S;QAAS,a;;QAZA,U;QAAA,SAaqB,MAbf,OAA  
N,GAAa,CAAb,I;QAAb,aAAU,CAAV,mB;UAakC,MAZ9B,CAAM,CAAN,IAyS,C,IAZ3B,CAAK,CAAL,C;;QAY  
H,OAAzB,M;;MAHIC,W;K;2EAOJ,yB;MAAA,iC;MAAA,6B;QACoF,YAAa,aAAa,IAAb,EAAmB,KAAAnB,C;QA  
IBhF,Q;QAAA,OAAA,KAAM,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,iB;UACI,MAAM,CAAN,IAiBoH,IAj  
BzG,CAAK,CAAL,C;;QAIbIE,OAfzE,K;O;KAcX,C;IAGA,+B;MAKiB,IAAN,I;MAFP,aAAa,IAAb,WAAa,CAA  
D,IAAC,C;MACb,gBAAkB,W;MAEd,IADS,IACT,mBADs,IACT,EAAM,IAAN,YADS,IACT,EAAY,KAAZ,E;Q  
AAqB,a;;QA1BZ,U;QAAA,SA2BkB,MA3BZ,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,mB;UA2B+B,MA1B3  
B,CAAM,CAAN,IA0BmC,IA1BxB,CAAK,CAAL,C;;QA0BH,OAAmB,M;;MAF/B,W;K;qEAMJ,yB;MAAA,2B;  
MAAA,gC;MAAA,6B;QAGiB,Q;QADb,YAAY,UAAU,IAAV,EAAgB,IAAhB,C;QACC,OAAA,KAAM,OAAN,G  
AAa,CAAb,I;QAAb,aAAU,CAAV,iB;UACI,YACY,eAAK,CAAL,E;UACpB,KAAK,CAAC,CAAD,CAAG,GAA  
G,K;;QAEP,OAAO,K;O;KARX,C;mFAWA,yB;MAAA,mB;MAAA,gC;MAAA,6B;QAGiB,Q;QADb,YAAY,QA  
AY,IAAZ,C;QACC,OAAA,KAAM,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,iB;UACI,YACY,eAAK,CAAL,E;  
UACpB,KAAK,CAAC,CAAD,CAAG,GAAG,K;;QAEP,OAAO,K;O;KARX,C;IAWA,+B;MAIiB,IAAN,I;MAFP,a  
AAsB,MAAY,IAAZ,C;MACtB,gBAAkB,W;MAEd,IADS,IACT,mBADs,IACT,EAAM,IAAN,E;QAAC,oBAAa,M  
AAb,K;WACd,WAFS,IAET,S;QAAS,a;;QA3DA,U;QAAA,SA4DkB,MA5DZ,OAAN,GAAa,CAAb,I;QAAb,aAA  
U,CAAV,mB;UA4D+B,MA3D3B,CAAM,CAAN,IA2DmC,IA3DxB,CAAK,CAAL,C;;QA2DH,OAAmB,M;;MAH  
/B,W;K;qEAOJ,yB;MAAA,2B;MAAA,6B;QAC2E,YAAa,UAAU,IAAV,EAAgB,KAAbB,C;QAjEvE,Q;QAAA,O  
AAA,KAAM,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,iB;UACI,MAAM,CAAN,IAgEwG,IAhE7F,CAAK,CA  
AL,C;;QAgEwD,OA9DhE,K;O;KA6DX,C;IAGA,wC;MACiB,Q;MAAA,OAAA,KAAM,OAAN,GAAa,CAAb,I;M  
AAb,aAAU,CAAV,iB;QACI,MAAM,CAAN,IAAW,S;;MAEf,OAAO,K;K;IEIFX,iC;MAAA,qC;MAEI,iBAC8B,Q  
;MAE9B,iBAC8B,sB;MAE9B,yBAEsC,MAAM,G;MAE5C,yBAEsC,CAAC,GAAD,GAAO,G;MAE7C,WAEwB,  
EAAE,MAAM,GAAR,C;MAExB,kBACuB,C;MAEvB,iBACsB,E;K;;IAxB1B,6C;MAAA,4C;QAAA,2B;;MAAA,  
qC;K;IA2BA,gC;MAAA,oC;MAEI,iBAC6B,O;MAE7B,iBAC6B,Y;MAE7B,yBAEqC,MAAO,G;MAE5C,yBAEq  
C,CAAC,GAAD,GAAQ,G;MAE7C,WAEuB,EAAE,MAAO,GAAT,C;MAEvB,kBACuB,C;MAEvB,iBACsB,E;K;;  
;IAxB1B,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;IA2BA,8B;MAAA,kC;MAEI,iBACqB,W;MAErB,iBACqB,U;  
MAErB,kBACuB,C;MAEvB,iBACsB,E;K;;IAZ1B,0C;MAAA,yC;QAAA,wB;;MAAA,kC;K;IAeA,+B;MAAA,m  
C;MAEI,iBACJ,MAAM,KAAoB,U;MAEtB,iBACJ,MAAM,KAAoB,U;MAEtB,kBACuB,C;MAEvB,iBACsB,E;K;  
;;IAZ1B,2C;MAAA,0C;QAAA,yB;;MAAA,mC;K;IAeA,gC;MAAA,oC;MAEI,iBACuB,U;MAEvB,iBACuB,K;M  
AEvB,kBACuB,C;MAEvB,iBACsB,E;K;;IAZ1B,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;IAeA,+B;MAAA,mC;

MAEI,iBACsB,Q;MAEtB,iBACsB,G;MAEtB,kBACuB,C;MAEvB,iBACsB,C;K;;;IAZ1B,2C;MAAA,0C;QAAA,y  
B;;MAAA,mC;K;IAeA,+B;MAAA,mC;MAEI,iBACmC,C;MAEnC,iBACmC,K;MAEnC,0BAC4C,K;MAE5C,0B  
AC4C,K;MAE5C,yBAC2C,K;MAE3C,yBAC2C,K;MAE3C,qBACuC,uB;MAEvC,qBACuC,sB;MAEvC,kBACuB,  
C;MAEvB,iBACsB,E;K;;;IA9B1B,2C;MAAA,0C;QAAA,yB;;MAAA,mC;K;IAiCA,iC;MAAA,qC;K;;;IAAA,6C;  
MAAA,4C;QAAA,2B;;MAAA,qC;K;IAEA,kC;MAAA,sC;K;;;IAAA,8C;MAAA,6C;QAAA,4B;;MAAA,sC;K;;;;  
;;aCu5vBoB,gB;;cC9puB0C,mB;;gBAqRvC,yB;eAAYB,wB;;uBAgBzB,gC;sBAAwB,+B;mC  
A4JjC,qB;mCA5ImC,qB;;kBAQ1B,2B;iBAA0B,0B;;eC74BgB,wB;sBCoBA,sB;iBCnBA,0B;;gCC  
9SIB,yC;+BCVA,uC;+BCAA,sC;;aC+EgD,e;gCC0E/E,+B;+BAIW,sC;gCCgxCc,+B;0BAHvB,kC;uBAr6BO,gC;y  
BA8WD,iC;0BACA,mC;yBA4JA,iC;gCAmZP,oC;+BAbc,oC;+BAEC,+B;yBAEQ,kC;;gBCh1C6C,yB;;;  
;;ICtErF,yB;K;;IAQA,6B;K;;IAUA,oC;K;;IC3BA,kD;MAMuF,wC;K;IANvF,4CAOI,Y;MAAuC,8B;K;I  
AP3C,8E;ICGA,kD;MAQuF,wC;K;IARvF,4CASI,Y;MAAuC,8B;K;IAT3C,8E;0FdOA,qB;MAQI,OAAO,UAAI,C  
AAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;M  
AQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CA  
AJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MA  
QI,OAAO,UAAI,CAAJ,C;K;0FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,  
C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,  
OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;  
K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;0FAGX,qB;MAQI,O  
AAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;  
4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OA  
AO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4  
FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;0FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAA  
O,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4F  
AGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,  
UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;0FAG  
X,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UA  
AI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,  
qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAA  
I,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;IAGX,sC;MAII,OAAO,mBAAQ,OAAR,KAAoB,C;K;I  
AG/B,wC;MAII,OAAO,qBAAQ,OAAR,KAAoB,C;K;IAG/B,wC;MAII,OAAO,qBAAQ,OAAR,KAAoB,C;K;IAG/  
B,wC;MAII,OAAO,qBAAQ,OAAR,KAAoB,C;K;IAG/B,wC;MAII,OAAO,qBAAQ,OAAR,KAAoB,C;K;IAG/B,w  
C;MAMW,c;;QA06XS,Q;QAaHb,iD;UAAgB,gBAaHb,e;UAAsB,IAAc,SA16XvB,YA06XS,C;YAAwB,aAAO,I;  
YAAP,e;;;QAC9C,aAAO,K;;;MA36XP,iB;K;IAGJ,wC;MAMW,c;;QA26XS,Q;QAaHb,iD;UAAgB,gBAaHb,e;U  
AAsB,IAAc,SA36XvB,YA26XS,C;YAAwB,aAAO,I;YAAP,e;;;QAC9C,aAAO,K;;;MA56XP,iB;K;IAGJ,wC;MAII  
,OAAO,qBAAQ,OAAR,KAAoB,C;K;IAG/B,wC;MAII,OAAO,qBAAQ,OAAR,KAAoB,C;K;oGAKe/B,yB;MAA  
A,8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAc,UAAI,KAAJ,CAAtC,GAAcD,aAAa  
,KAAb,C;O;KAPjE,C;sGAUA,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GA  
AsC,UAAI,KAAJ,CAAtC,GAAcD,aAAa,KAAb,C;O;KAPjE,C;sGAUA,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,  
SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAc,UAAI,KAAJ,CAAtC,GAAcD,aAAa,KAAb,C;O;KAPjE,C;sGAUA,y  
B;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAc,UAAI,KAAJ,CAAtC,GAAc  
D,aAAa,KAAb,C;O;KAPjE,C;sGAUA,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBA  
A3B,GAAc,UAAI,KAAJ,CAAtC,GAAcD,aAAa,KAAb,C;O;KAPjE,C;sGAUA,yB;MAAA,8D;MAAA,iD;QAOI,  
OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAc,UAAI,KAAJ,CAAtC,GAAcD,aAAa,KAAb,C;O;KAPjE,C;s  
GAUA,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAc,UAAI,KAAJ,CAA  
tC,GAAcD,aAAa,KAAb,C;O;KAPjE,C;sGAUA,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SA  
AS,wBAA3B,GAAc,UAAI,KAAJ,CAAtC,GAAcD,aAAa,KAAb,C;O;KAPjE,C;sGAUA,yB;MAAA,8D;MAAA,g  
C;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAc,UAAI,KAAJ,CAAtC,GAAcD,uBAAa,

KAAb,E;O;KAPjE,C;oGAUA,yB;MAAA,sD;MAAA,mC;QAOI,OAA Y,UAAL,SAAK,EAAU,KAAV,C;O;KAPh B,C;qGAUA,yB;MAAA,qD;MAAA,mC;QAOI,OAA Y,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;sGAUA,yB; MAAA,sD;MAAA,mC;QAOI,OAA Y,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;sGAUA,yB;MAAA,sD;MAA A,mC;QAOI,OAA Y,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;sGAUA,yB;MAAA,sD;MAAA,mC;QAOI,OAA Y,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;sGAUA,yB;MAAA,sD;MAAA,mC;QAOI,OAA Y,UAAL,SAAK,E AAU,KAAV,C;O;KAPhB,C;sGAUA,yB;MAAA,sD;MAAA,mC;QAOI,OAA Y,UAAL,SAAK,EAAU,KAAV,C;O; KAPhB,C;sGAUA,yB;MAAA,sD;MAAA,mC;QAOI,OAA Y,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;sGAUA ,yB;MAAA,sD;MAAA,mC;QAOI,OAA Y,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;8EAUa,gC;MAOW,sB;;Q AkcS,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IAIcH,SAkcO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP, uB;;;QAC9C,qBAAO,I;;;MAncP,yB;K;gFAGJ,gC;MAOW,sB;;QAgcS,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,I AhcH,SAgcO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MAjcP,yB;K;gFAGJ,g C;MAOW,sB;;QA8bS,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IA9bH,SA8bO,CAAU,OAAV,CAAJ,C;YAAwB,q BAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MA/bP,yB;K;gFAGJ,gC;MAOW,sB;;QA4bS,Q;QAaHb,iD;UAAgB,cA AhB,e;UAAsB,IA5bH,SA4bO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MA7b P,yB;K;gFAGJ,gC;MAOW,sB;;QA0bS,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IA1bH,SA0bO,CAAU,OAAV,C AAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MA3bP,yB;K;gFAGJ,gC;MAOW,sB;;QAwbS,Q;QA Ahb,iD;UAAgB,cAAhB,e;UAAsB,IAxbH,SAwbO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC 9C,qBAAO,I;;;MAzbP,yB;K;gFAGJ,gC;MAOW,sB;;QAsbS,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IAtbH,SAsb O,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MAvbP,yB;K;gFAGJ,gC;MAOW, sB;;QAobS,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IApbH,SAobO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O; YAAP,uB;;;QAC9C,qBAAO,I;;;MArbP,yB;K;gFAGJ,yB;MAqbA,oC;MAAA,gC;MArBA,uC;QAOW,sB;;UAkbS, Q;UAAhB,iD;YAAgB,cAAhB,0B;YAAsB,IAIbH,SAkbO,CAAU,oBAAV,CAAJ,C;cAAwB,qBAAO,O;cAAP,uB;; ;UAC9C,qBAAO,I;;;QAnbP,yB;O;KAPJ,C;sFAUA,yB;MAi2CA,0D;MAAA,+C;MAj2CA,uC;QAOW,qB;;UAg2C O,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAAAd,OAAC,cAAAd,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YA Cd,IAI2Cc,SAk2CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAp2CP,wB;O;K APJ,C;wFAUA,yB;MAo2CA,0D;MAAA,+C;MAp2CA,uC;QAOW,qB;;UAm2CO,Q;UAAA,OAAa,SAAR,sBAA Q,CAAb,W;UAAAd,OAAC,cAAAd,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IAr2Cc,SAq2CV,CAAU,OAA V,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAv2CP,wB;O;KAPJ,C;wFAUA,yB;MAu2CA,0D; MAAA,+C;MAv2CA,uC;QAOW,qB;;UAs2CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAAAd,OAAC,cAAAd,C; YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IAx2Cc,SAw2CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cA AP,sB;;;UAE5B,oBAAO,I;;;QA12CP,wB;O;KAPJ,C;wFAUA,yB;MA02CA,0D;MAAA,+C;MA12CA,uC;QAOW, qB;;UAy2CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAAAd,OAAC,cAAAd,C;YAAc,uB;YACV,cAAc,UAAK,K AAL,C;YACd,IA32Cc,SA22CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QA72 CP,wB;O;KAPJ,C;wFAUA,yB;MA62CA,0D;MAAA,+C;MA72CA,uC;QAOW,qB;;UA42CO,Q;UAAA,OAAa,SA AR,sBAAQ,CAAb,W;UAAAd,OAAC,cAAAd,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IA92Cc,SA82CV,C AAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAh3CP,wB;O;KAPJ,C;wFAUA,yB;M Ag3CA,0D;MAAA,+C;MAh3CA,uC;QAOW,qB;;UA+2CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAAAd,OA AC,cAAAd,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IAj3Cc,SAi3CV,CAAU,OAAV,CAAJ,C;cAAwB,oB AAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAn3CP,wB;O;KAPJ,C;wFAUA,yB;MAm3CA,0D;MAAA,+C;MAN3CA, uC;QAOW,qB;;UAk3CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAAAd,OAAC,cAAAd,C;YAAc,uB;YACV,cAA c,UAAK,KAAL,C;YACd,IAp3Cc,SAo3CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAA O,I;;;QAt3CP,wB;O;KAPJ,C;wFAUA,yB;MAs3CA,0D;MAAA,+C;MAT3CA,uC;QAOW,qB;;UAq3CO,Q;UAAA, OAAa,SAAR,sBAAQ,CAAb,W;UAAAd,OAAC,cAAAd,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IAv3Cc,S Au3CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAz3CP,wB;O;KAPJ,C;wFA UA,yB;MAy3CA,0D;MAAA,+C;MAAA,oC;MAz3CA,uC;QAOW,qB;;UAw3CO,Q;UAAA,OAAa,SAAR,sBAAQ ,CAAb,W;UAAAd,OAAC,cAAAd,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IA13Cc,SA03CV,CAAU,oBAA V,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QA53CP,wB;O;KAPJ,C;IAUA,0B;MAMI,IAovNO, qBAAQ,CApvNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAAL,C;K;IAGX,4B;MAMI,IAiv



gC;MAAA,uC;QAIoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UAAsB,IAAI,UAAU ,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,OAAO,I;O;KALX,C;wFAQA,yB;MAAA,8D;MAAA,iD;QAKI,OA AW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAcC,UAAI,KA AJ,CAAtC,GAAcD,aAAa,KAAb,C;O;KALjE,C;0FA QA,yB;MAAA,8D;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAcC,UAAI,KA AJ,CAAtC, GAAcD,aAAa,KAAb,C;O;KALjE,C;0FAQA,yB;MAAA,8D;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS, wBAA3B,GAAcC,UAAI,KA AJ,CAAtC,GAAcD,aAAa,KAAb,C;O;KALjE,C;0FAQA,yB;MAAA,8D;MAAA,iD;Q AKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAcC,UAAI,KA AJ,CAAtC,GAAcD,aAAa,KAAb,C;O;KALj E,C;0FAQA,yB;MAAA,8D;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAcC,UAAI,KA AJ, CAAtC,GAAcD,aAAa,KAAb,C;O;KALjE,C;0FAQA,yB;MAAA,8D;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAA c,SAAS,wBAA3B,GAAcC,UAAI,KA AJ,CAAtC,GAAcD,aAAa,KAAb,C;O;KALjE,C;0FAQA,yB;MAAA,8D;MA AA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAcC,UAAI,KA AJ,CAAtC,GAAcD,aAAa,KAAb,C; O;KALjE,C;0FAQA,yB;MAAA,8D;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAcC,UAA I,KA AJ,CAAtC,GAAcD,aAAa,KAAb,C;O;KALjE,C;0FAQA,yB;MAAA,8D;MAAA,gC;MAAA,iD;QAKI,OAAW ,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAcC,UAAI,KA AJ,CAAtC,GAAcD,uBAAa,KAAb,E;O;KALjE,C;IAQA, qC;MAMI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAcC,UAAI,KA AJ,CAAtC,GAAcD,I;K;IAGjE,uC;MA MI,OAAW,SAAS,CAAT,IAAc,SAAS,0BAA3B,GAAcC,UAAI,KA AJ,CAAtC,GAAcD,I;K;IAGjE,uC;MAMI,OA AW,SAAS,CAAT,IAAc,SAAS,0BAA3B,GAAcC,UAAI,KA AJ,CAAtC,GAAcD,I;K;IAGjE,uC;MAMI,OAAW,SA AS,CAAT,IAAc,SAAS,0BAA3B,GAAcC,UAAI,KA AJ,CAAtC,GAAcD,I;K;IAGjE,uC;MAMI,OAAW,SAAS,CA AT,IAAc,SAAS,0BAA3B,GAAcC,UAAI,KA AJ,CAAtC,GAAcD,I;K;IAGjE,uC;MAMI,OAAW,SAAS,CAAT,IAA c,SAAS,0BAA3B,GAAcC,UAAI,KA AJ,CAAtC,GAAcD,I;K;IAGjE,uC;MAMI,OAAW,SAAS,CAAT,IAAc,SAAS ,0BAA3B,GAAcC,UAAI,KA AJ,CAAtC,GAAcD,I;K;IAGjE,uC;MAMI,OAAW,SAAS,CAAT,IAAc,SAAS,0BAA3 B,GAAcC,UAAI,KA AJ,CAAtC,GAAcD,I;K;IAGjE,uC;MAMI,OAAW,SAAS,CAAT,IAAc,SAAS,0BAA3B,GAA cC,UAAI,KA AJ,CAAtC,GAAcD,I;K;IAGjE,qC;MAIL,IAAI,eAAJ,C;QACI,wD;UACI,IAAI,UAAK,KAAL,SAAJ, C;YACI,OAAO,K;;;QAIIf,8D;UACI,IAAI,gBAAW,UAAK,OAAL,CAAX,CAAJ,C;YACI,OAAO,O;;;MAlnB,OA AO,E;K;IAGX,uC;MAIL,wD;QACI,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IA GX,uC;MAIL,wD;QACI,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,uC;MA II,wD;QACI,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,uC;MAIL,wD;QAC I,IAAI,gBAAW,UAAK,KAAL,CAAX,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,uC;MAMI,wD;QACI, IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,uC;MAMI,wD;QACI,IAAI,YA AW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,uC;MAIL,wD;QACI,IAAI,YAAW,UAAK ,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,uC;MAIL,wD;QACI,IAAI,YAAW,UAAK,KAAL,CA Af,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;8FAGX,gC;MAIL,wD;QACI,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAIL,wD;QACI,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C ;UACI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAIL,wD;QACI,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;U ACI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAIL,wD;QACI,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;UA CI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAIL,wD;QACI,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;UACI, OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAIL,wD;QACI,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;UACI,OA AO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAIL,wD;QACI,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;UACI,OA AO,K;;;MAGf,OAAO,E;K;gGAGX,yB;MAAA,oC;MAAA,uC;QAIIf,wD;UACI,IAAI,UAAU,sBAAK,KAAL,EA AV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAIkB,Q; QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAC,cAAAd,C;UAAc,uB;UACV,IAAI,UAAU,UAAK,KAAL,CA AV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAIkB ,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAC,cAAAd,C;UAAc,uB;UACV,IAAI,UAAU,UAAK,KAAL, CA AV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QA I kB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAC,cAAAd,C;UAAc,uB;UACV,IAAI,UAAU,UAAK,KA AL,CA AV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,0D;MAAA,+C;MAAA,uC; QA I kB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAC,cAAAd,C;UAAc,uB;UACV,IAAI,UAAU,UAAK,

KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAC,cAAd,C;UAAc,uB;UACV,IAAI,UAAU,UA AK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,0D;MAAA,+C;MA AA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAC,cAAd,C;UAAc,uB;UACV,IAAI,UAAU, UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,0D;MAAA,+C; MAAA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAC,cAAd,C;UAAc,uB;UACV,IAAI,UA AU,UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,0D;MAAA ,+C;MAAA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAC,cAAd,C;UAAc,uB;UACV,IAAI, UAAU,UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,0D;MA AA,+C;MAAA,oC;MAAA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAC,cAAd,C;UAAc,u B;UACV,IAAI,UAAU,sBAAK,KAAL,EAHV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;IAYA,yB; MAQI,IAg7LO,qBAAQ,CAh7Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,wBAAL,C;K;IAG X,2B;MAQI,IA26LO,qBAAQ,CA36Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,0BAAL,C;K ;IAGX,2B;MAQI,IA56LO,qBAAQ,CAt6Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,0BAAL, C;K;IAGX,2B;MAQI,IAi6LO,qBAAQ,CAj6Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,0BA AL,C;K;IAGX,2B;MAQI,IA45LO,qBAAQ,CA55Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK ,0BAAL,C;K;IAGX,2B;MAQI,IAu5LO,qBAAQ,CAv5Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,U AAK,0BAAL,C;K;IAGX,2B;MAQI,IAk5LO,qBAAQ,CAi5Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OA AO,UAAK,0BAAL,C;K;IAGX,2B;MAQI,IA64LO,qBAAQ,CA74Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MAC V,OAAO,UAAK,0BAAL,C;K;IAGX,2B;MAQI,IAw4LO,qBAAQ,CAx4Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C; MACV,OAAO,UAAK,0BAAL,C;K;gFAGX,yB;MAAA,0D;MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA, OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAC,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;U ACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEEnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;gFAe A,yB;MAAA,0D;MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb, W;QAAd,OAAC,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB, OAAO,O;;QAEEnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;iFAeA,yB;MAAA,0D;MAAA,+C;MAAA,iE;MAA A,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAC,cAAd,C;UAAc,uB;UACV,c AAAC,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEEnC,MAAM,gCAAuB,mDAA vB,C;O;KAZV,C;iFAeA,yB;MAAA,0D;MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAA L,SAAK,CAAQ,CAAb,W;QAAd,OAAC,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,O AAV,CAAJ,C;YAAwB,OAAO,O;;QAEEnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;iFAeA,yB;MAAA,0D;MA AA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAC,cAA Ad,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEEnC,M AAM,gCAAuB,mDAAvB,C;O;KAZV,C;iFAeA,yB;MAAA,0D;MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QA AA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAC,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C ;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEEnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;iFA eA,yB;MAAA,0D;MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAA b,W;QAAd,OAAC,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAw B,OAAO,O;;QAEEnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;iFAeA,yB;MAAA,0D;MAAA,+C;MAAA,iE;MA AA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAC,cAAd,C;UAAc,uB;UACV, cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEEnC,MAAM,gCAAuB,mDA AvB,C;O;KAZV,C;iFAeA,yB;MAAA,0D;MAAA,+C;MAAA,oC;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa, SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAC,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IA AI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QAEEnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;IAeA,yC;MA KsB,UAMA,M;MAPIB,IAAI,eAAJ,C;QACkB,OAAQ,WAAR,sBAAQ,CAAR,W;QAAd,OAAC,cAAd,C;UAAc,uB ;UACV,IAAI,UAAK,KAAL,SAAJ,C;YACI,OAAO,K;;;QAID,SAAQ,WAAR,sBAAQ,CAAR,W;QAAd,OAAC,g BAAd,C;UAAc,2B;UACV,IAAI,gBAAW,UAAK,OAAL,CAAX,CAAJ,C;YACI,OAAO,O;;;MAInB,OAAO,E;K;I AGX,2C;MAIkB,Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,W;MAAd,OAAC,cAAd,C;QAAC,uB;QACV,IAAI,YA

AW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,2C;MAIkB,Q;MAAA,OAAQ,WAAR,wB  
AAQ,CAAR,W;MAAd,OAAc,cAAd,C;QAAC,uB;QACV,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;  
MAGf,OAAO,E;K;IAGX,2C;MAIkB,Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,W;MAAd,OAAc,cAAd,C;QAAC,  
uB;QACV,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,2C;MAIkB,Q;MAA  
A,OAAQ,WAAR,wBAAQ,CAAR,W;MAAd,OAAc,cAAd,C;QAAC,uB;QACV,IAAI,gBAAW,UAAK,KAAL,CA  
AX,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,2C;MAMkB,Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,  
W;MAAd,OAAc,cAAd,C;QAAC,uB;QACV,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAA  
O,E;K;IAGX,2C;MAMkB,Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,W;MAAd,OAAc,cAAd,C;QAAC,uB;QACV,  
IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,2C;MAIkB,Q;MAAA,OAAQ,W  
AAR,wBAAQ,CAAR,W;MAAd,OAAc,cAAd,C;QAAC,uB;QACV,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,O  
AAO,K;;;MAGf,OAAO,E;K;IAGX,2C;MAIkB,Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,W;MAAd,OAAc,cAAd  
,C;QAAC,uB;QACV,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,+B;MAMI,  
OA8jLO,qBAAQ,CA9jLR,GAAe,IAAf,GAAYB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OA6jLO,q  
BAAQ,CA7jLR,GAAe,IAAf,GAAYB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OA4jLO,qBAAQ,CA  
5jLR,GAAe,IAAf,GAAYB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OA2jLO,qBAAQ,CA3jLR,GAA  
e,IAAf,GAAYB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OA0jLO,qBAAQ,CA1jLR,GAAe,IAAf,GA  
AYB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OAyjLO,qBAAQ,CAzjLR,GAAe,IAAf,GAAYB,UAA  
K,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OAwjLO,qBAAQ,CAXjLR,GAAe,IAAf,GAAYB,UAAK,mBAAO  
,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OAujLO,qBAAQ,CAvjLR,GAAe,IAAf,GAAYB,UAAK,mBAAO,CAAP,IA  
AL,C;K;IAGpC,iC;MAMI,OAsjLO,qBAAQ,CAtjLR,GAAe,IAAf,GAAYB,UAAK,mBAAO,CAAP,IAAL,C;K;4F  
AGpC,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;Q  
AAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OA  
AO,O;;QAEnC,OAAO,I;O;KAVX,C;4FAaA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SA  
AR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,  
UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;6FAaA,yB;MAAA,0D;MAAA,+C;MA  
AA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,  
cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;6F  
AaA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QA  
Ad,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAA  
O,O;;QAEnC,OAAO,I;O;KAVX,C;6FAaA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SA  
AR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,U  
AAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;6FAaA,yB;MAAA,0D;MAAA,+C;MAA  
A,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,c  
AAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;6FA  
aA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd  
,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O  
;;QAEnC,OAAO,I;O;KAVX,C;6FAaA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,Y  
AAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAA  
U,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;6FAaA,yB;MAAA,0D;MAAA,+C;MAAA,o  
C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;  
UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KA  
VX,C;kFAaA,yB;MAAA,mC;MAAA,gD;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;oFAWA,yB;MA  
AA,mC;MAAA,gD;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;oFAWA,yB;MAAA,mC;MAAA,gD;  
MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;oFAWA,yB;MAAA,mC;MAAA,gD;MAAA,4B;QAQI,O  
AAO,kBAAO,cAAP,C;O;KARX,C;oFAWA,yB;MAAA,mC;MAAA,gD;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,  
C;O;KARX,C;oFAWA,yB;MAAA,mC;MAAA,gD;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;oFAW  
A,yB;MAAA,mC;MAAA,gD;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;oFAWA,yB;MAAA,mC;M  
AAA,gD;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;oFAWA,yB;MAAA,mC;MAAA,gD;MAAA,4B;





A,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAoB,I;QACpB,YAAY,K;QACZ,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KA AJ,C;cAA W,MAAM,8BAAYB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAA Y,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,2D;O;KafX,C;qFakBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC ;QAMoB,UAST,M;QAXP,aAAqB,I;QACrB,YAAY,K;QACZ,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;U ACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KA AJ,C;cAAW,MAAM,8BAAYB,gDAAzB,C;YACjB,SAAS,O;Y ACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAA Y,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,2D;O;Kaf X,C;qFakBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAmB,I;QACnB,YAA Y,K;QACZ,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KA AJ,C;cAAW,MAAM,8BAAYB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UA AY,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,2D;O;Kaf X,C;qFakBA,yB;MAAA,kF;MAAA,iE;MAAA,8B; MAAA,uC;QAMoB,UAST,M;QAXP,aAAoB,I;QACpB,YAAY,K;QACZ,wBAAGB,SAAhB,gB;UAAgB,cAAA,S AAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KA AJ,C;cAAW,MAAM,8BAAYB,gDAAzB,C;YACjB, SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAA Y,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,i E;O;KafX,C;qFakBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAqB,I;QACr B,YAAY,K;QACZ,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,I AAI,KA AJ,C;cAAW,MAAM,8BAAYB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KA AL,C;UAA Y,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,2D;O;KafX,C;qFakBA,yB;MAAA,kF;MAAA,iE;M AAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAA sB,I;QACtB,YAAY,K;QACZ,wBAAGB,SAAhB,gB;UAAgB,c AAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KA AJ,C;cAAW,MAAM,8BAAYB,gDAAzB,C; YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAA Y,MAAM,gCAAuB,mDAAvB,C;QAEIB, OAAO,2D;O;KafX,C;qFakBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAu B,I;QACvB,YAAY,K;QACZ,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ, C;YACI,IAAI,KA AJ,C;cAAW,MAAM,8BAAYB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,C AAC,KAAL,C;UAA Y,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,4D;O;KafX,C;qFakBA,yB;MAAA,oC;MA AA,kF;MAAA,gC;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAoB,I;QACpB,YAAY,K;QAC Z,wBAAGB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,IAAI,UAAU,oBAAV,CAAJ,C;YACI,IAAI,K AAJ,C;cAAW,MAAM,8BAAYB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;U AAY,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,4E;O;KafX,C;IAkBA,iC;MAII,OAAW,qBAAQ,CAAZ,GAA e,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4 B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAA W,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UA AK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K; IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qB AAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,C AAL,CAAf,GAA4B,I;K;gGAGvC,gC;MAMoB,Q;MAFhB,aAAiB,I;MACjB,YAAY,K;MACZ,wBAAGB,SAAhB, gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KA AJ,C;YAAW,OAAO,I;UACIB, SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAA Y,OAAO,I;MACnB,OAAO,M;K;gGAGX,gC;MA MoB,Q;MAFhB,aAAoB,I;MACpB,YAAY,K;MACZ,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI ,UAAU,OAAV,CAAJ,C;UACI,IAAI,KA AJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI, CAAC,KAAL,C;QAA Y,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aAAqB,I;MACrB,YAAY ,K;MACZ,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KA AJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAA Y,OAAO,I;MACnB, OAAO,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aAAmB,I;MACnB,YAAY,K;MACZ,wBAAGB,SAAhB,gB;QAAgB, cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KA AJ,C;YAAW,OAAO,I;UACIB,SAAS,O;U ACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAA Y,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MAMoB,Q;M AFhB,aAAoB,I;MACpB,YAAY,K;MACZ,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,UAAU,O AAV,CAAJ,C;UACI,IAAI,KA AJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,K

AAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aAAqB,I;MACrB,YAAY,K;MACZ,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aAAqB,I;MACrB,YAAY,K;MACZ,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aAAqB,I;MACrB,YAAY,K;MACZ,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,yB;MAAA,oC;MAAA,gC;MAAA,uC;QAMoB,Q;QAFhB,aAAoB,I;QACpB,YAAY,K;QACZ,wBAAgB,SAAhB,gB;UAGB,cAAhB,UAGB,SAAhB,O;UACI,IAAI,UAAU,oBAAV,CAAJ,C;YACI,IAAI,KAAJ,C;CAAW,OAAO,I;YACIB,SAAS,O;YACT,QAAQ,I;QAGhB,IAAI,CAAC,KAAL,C;UAAAY,OAAO,I;QACnB,OAAO,M;O;KADx,C;IAiBA,4B;Me9qGI,IAAI,EfsrGI,KAAK,CetrGT,CAAJ,C;QACI,cfqrGc,sD;QeprGd,MAAM,gCAAYB,OAAQ,WAAjC,C;MfqrGV,OAAO,oBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;Me1rGI,IAAI,EfksGI,KAAK,CelsGT,CAAJ,C;QACI,cfisGc,sD;QehsGd,MAAM,gCAAYB,OAAQ,WAAjC,C;MfisGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;MetsGI,IAAI,Ef8sGI,KAAK,Ce9sGT,CAAJ,C;QACI,cf6sGc,sD;Qe5sGd,MAAM,gCAAYB,OAAQ,WAAjC,C;Mf6sGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;MeltGI,IAAI,Ef0tGI,KAAK,Ce1tGT,CAAJ,C;QACI,cfytGc,sD;QextGd,MAAM,gCAAYB,OAAQ,WAAjC,C;MfytGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;Me9tGI,IAAI,EfsuGI,KAAK,CetuGT,CAAJ,C;QACI,cfquGc,sD;QepuGd,MAAM,gCAAYB,OAAQ,WAAjC,C;MfquGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;Me1uGI,IAAI,EfkuGI,KAAK,CelvGT,CAAJ,C;QACI,cfivGc,sD;QehvGd,MAAM,gCAAYB,OAAQ,WAAjC,C;MfivGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;MetvGI,IAAI,Ef8vGI,KAAK,Ce9vGT,CAAJ,C;QACI,cf6vGc,sD;Qe5vGd,MAAM,gCAAYB,OAAQ,WAAjC,C;Mf6vGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;MelwGI,IAAI,Ef0wGI,KAAK,Ce1wGT,CAAJ,C;QACI,cfywGc,sD;QexwGd,MAAM,gCAAYB,OAAQ,WAAjC,C;MfywGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;Me9wGI,IAAI,EfsxGI,KAAK,CetxGT,CAAJ,C;QACI,cfqxGc,sD;QepxGd,MAAM,gCAAYB,OAAQ,WAAjC,C;MfqxGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,gC;Me1xGI,IAAI,EfkyGI,KAAK,CelyGT,CAAJ,C;QACI,cfiyGc,sD;QehyGd,MAAM,gCAAYB,OAAQ,WAAjC,C;MfiyGV,OAAO,gBAAgB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;MetyGI,IAAI,Ef8yGI,KAAK,Ce9yGT,CAAJ,C;QACI,cf6yGc,sD;Qe5yGd,MAAM,gCAAYB,OAAQ,WAAjC,C;Mf6yGV,OAAO,kBAAgB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;MelzGI,IAAI,Ef0zGI,KAAK,Ce1zGT,CAAJ,C;QACI,cfyzGc,sD;QexzGd,MAAM,gCAAYB,OAAQ,WAAjC,C;MfyzGV,OAAO,kBAAgB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;Me9zGI,IAAI,Efs0GI,KAAK,Ce0GT,CAAJ,C;QACI,cfq0Gc,sD;Qep0Gd,MAAM,gCAAYB,OAAQ,WAAjC,C;Mfq0GV,OAAO,kBAAgB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;Me10GI,IAAI,Efk1GI,KAAK,Ce11GT,CAAJ,C;QACI,cfi1Gc,sD;Qeh1Gd,MAAM,gCAAYB,OAAQ,WAAjC,C;Mfi1GV,OAAO,kBAAgB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;Met1GI,IAAI,Ef81GI,KAAK,Ce91GT,CAAJ,C;QACI,cf61Gc,sD;Qe51Gd,MAAM,gCAAYB,OAAQ,WAAjC,C;Mf61GV,OAAO,kBAAgB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;Mel2GI,IAAI,Ef02GI,KAAK,Ce12GT,CAAJ,C;QACI,cfy2Gc,sD;Qex2Gd,MAAM,gCAAYB,OAAQ,WAAjC,C;Mfy2GV,OAAO,kBAAgB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;Me92GI,IAAI,Efs3GI,KAAK,Cet3GT,CAAJ,C;QACI,cfq3Gc,sD;Qep3Gd,MAAM,gCAAYB,OAAQ,WAAjC,C;Mfq3GV,OAAO,kBAAgB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;Me13GI,IAAI,Efk4GI,KAAK,Ce14GT,CAAJ,C;QACI,cfi4Gc,sD;Qeh4Gd,MAAM,gCAAYB,OAAQ,WAAjC,C;Mfi4GV,OAAO,kBAAgB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;gGAGX,yB;MAAA,8D;MAAA,4C;MAAA,qD;MAAA,uC;QAMI,iBAAC,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;QAGf,OAAO,W;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,2C;MAAA,qD;MAAA,uC;QAMI,iBAAC,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,g



W,kBAAS,gB;QAugBA,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IAvgBa,SAugBT,CAAU,OAAV,CAAJ,C;YAAwB,WAAY,WAAI,OAAJ,C;;QAvGB1D,OAwgBO,W;O;KA9gBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QAwgBA,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IAxgBc,SAwgBV,CAAU,OAAV,CAAJ,C;YAAwB,WAAY,WAAI,OAAJ,C;;QAxgB1D,OAYgBO,W;O;KA/gBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QAYgBA,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IAzgBe,SAygBX,CAAU,OAAV,CAAJ,C;YAAwB,WAAY,WAAI,OAAJ,C;;QAZgB1D,OA0gBO,W;O;KAhhBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QA0gBA,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IA1gBgB,SA0gBZ,CAAU,OAAV,CAAJ,C;YAAwB,WAAY,WAAI,OAAJ,C;;QA1gB1D,OA2gBO,W;O;KAjhBX,C;oFASA,yB;MAAA,+D;MAAA,gC;MA3gBA,uC;QAMW,kBAAS,gB;QA2gBA,Q;QAaHb,iD;UAAgB,cAAhB,0B;UAAsB,IA3gBa,SA2gBT,CAAU,oBAAV,CAAJ,C;YAAwB,WAAY,WAAI,oBAAJ,C;;QA3gB1D,OA4gBO,W;O;KAlhBX,C;gGASA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QASgTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UA16SI,IApGmC,SAoG/B,EAk6SkB,cAl6SIB,EAk6SkB,sBA16SIB,Wak6S2B,IA16S3B,CAAJ,C;YAA2C,sBAk6SZ,IA16SY,C;;QApG/C,OASGO,W;O;KA9GX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QAqgTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UA95SI,IAvGsC,SAuGIC,EA85SkB,cA95SIB,EA85SkB,sBA95SIB,WA85S2B,IA95S3B,CAAJ,C;YAA2C,sBA85SZ,IA95SY,C;;QAvG/C,OAYGO,W;O;KAjHX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QAogTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UA15SI,IA1GuC,SA0GnC,EA05SkB,cA15SIB,EA05SkB,sBA15SIB,WA05S2B,IA15S3B,CAAJ,C;YAA2C,sBA05SZ,IA15SY,C;;QA1G/C,OA4GO,W;O;KApHX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QAmgTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UA5SI,IA7GqC,SA6GjC,EA5S5SkB,cAt5SIB,EA5S5SkB,sBA5SIB,WAs5S2B,IA5S3B,CAAJ,C;YAA2C,sBA5SZ,IA5SY,C;;QA7G/C,OA+GO,W;O;KAvHX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QAkgTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UA15SI,IAhHsC,SAgHIC,EAk5SkB,cA15SIB,EAk5SkB,sBA15SIB,Wak5S2B,IA15S3B,CAAJ,C;YAA2C,sBAk5SZ,IA15SY,C;;QAhH/C,OAKHO,W;O;KA1HX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QAigTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UA94SI,IANHuC,SAmHnC,EA84SkB,cA94SIB,EA84SkB,sBA94SIB,WA84S2B,IA94S3B,CAAJ,C;YAA2C,sBA84SZ,IA94SY,C;;QAnH/C,OAqHO,W;O;KA7HX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QAgtTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UA14SI,IAthwC,SAsHpC,EA04SkB,cA14SIB,EA04SkB,sBA14SIB,WA04S2B,IA14S3B,CAAJ,C;YAA2C,sBA04SZ,IA14SY,C;;QAtH/C,OAwhO,W;O;KAhIX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QA+/SV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UA4SI,IAzHyC,SAyHrC,EA4S4SkB,cAt4SIB,EA4S4SkB,sBA4SIB,WAs4S2B,IA4S3B,CAAJ,C;YAA2C,sBA4SZ,IA4SY,C;;QAzH/C,OA2HO,W;O;KANIX,C;kGAWA,yB;MAAA,+D;MA2HA,gC;MAo4SA,oC;MA//SA,uC;QAQW,kBAAGB,gB;QA8/SV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,0B;UAAmB,eAAO,cAAP,EAAO,sBAAP,S;UAAA,cAAgB,iB;UA14S/B,IA5HsC,SA4HIC,CAAU,OAAV,EAAiB,OAAjB,CAAJ,C;YAA2C,sBAAI,OAAJ,C;;QA5H/C,OA8HO,W;O;KAIX,C;oGAWA,6C;MA26SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QA16SI,IAAI,Wak6SkB,cA16SIB,EAk6SkB,sBA16SIB,Wak6S2B,IA16S3B,CAAJ,C;UAA2C,sBAk6SZ,IA16SY,C;;MAE/C,OAAO,W;K;qGAGX,6C;MAu6SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QA95SI,IAAI,WA85SkB,cA95SIB,EA85SkB,sBA95SIB,WA85S2B,IA95S3B,CAAJ,C;UAA2C,sBA85SZ,IA95SY,C;;MAE/C,OAAO,W;K;sGAGX,6C;MAM6SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QA15SI,IAAI,WA05SkB,cA15SIB,EA05SkB,sBA15SIB,WA05S2B,IA15S3B,CAAJ,C;UAA2C,sBA05SZ,IA15SY,C;;MAE/C,OAAO,W;K;qGAGX,6C;MA+5SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QA5SI,IAAI,WAs5SkB,cAt5SIB,EA5S5SkB,sBA5SIB,WAs5S2B,IA5S3B,CAAJ,C;UAA2C,sBA5SZ,IA5SY,C;;MAE/C,OAAO,W;K;sGAGX,6C;MA25SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QA15SI,IAAI,Wak5SkB,cA15SIB,EAk5SkB,sBA15SIB,Wak5S2B,IA15S3B,CAAJ,C;UAA2C,sBAk5SZ,IA15SY,C;;MAE/C,OAAO,W;K;sGAGX,6C;MAu5SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QA94SI,IAAI,WA84SkB,cA94SIB,EA84SkB,sBA94SIB,WA84S2B,IA94S3B,CAAJ,C;UAA2C,sBA84SZ,IA94SY,C;;MAE/C,OAAO,W;K;sGAGX,6C;MAM5SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QA14SI,IAAI,WA04SkB,cA14SIB,EA04SkB,sBA14SIB,WA04S2B,IA14S3B,CAAJ,C;UAA2C,sBA04SZ,IA14SY,C;;MAE/C,OAAO,W;K;sGAGX,6C;MA+4SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QA4SI,IAAI,WAs4SkB,cAt4SIB,EA4S4SkB,sBA4SIB,WAs4S2B,IA4S3B,CAAJ,C;UAA2C,sBA4SZ,IA4SY,C;;MAE/C,OAAO,W;K;sGAGX,yB;MAAA,gC;MAo4SA,oC;MAP4SA,oD;QA24SiB,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,0B;UAAmB,e

AAO,cAAP,EAAO,sBAAP,S;UAAA,cAAgB,iB;UAI4S/B,IAAI,UAAU,OAAV,EAAiB,OAAjB,CAAJ,C;YAA2C,  
sBAAL,OAAJ,C;;QAE/C,OAAO,W;O;KAXX,C;sGAcA,yB;MAAA,+D;MAAA,sC;QAMW,kBAAmB,gB;QASV,  
Q;QAAhB,iD;UAAgB,cAAhB,e;UAAAsB,IAAI,YAAJ,C;YAAkB,WAAy,WAAI,OAAJ,C;;QATpD,OAuO,W;O;K  
AhBX,C;0GASA,4C;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,YAAJ,  
C;UAAkB,WAAy,WAAI,OAAJ,C;;MACpD,OAAO,W;K;wFAGX,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;  
B;QAoGH,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAAsB,IAAI,CAPGS,SAoGR,CAAU,OAAV,CAAL,C;YAAyB,WA  
AY,WAAI,OAAJ,C;;QApG3D,OAqGO,W;O;KA3GX,C;0FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;  
QAqGH,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAAsB,IAAI,CARGY,SAqGX,CAAU,OAAV,CAAL,C;YAAyB,WAAy  
,WAAI,OAAJ,C;;QArG3D,OAsgo,W;O;KA5GX,C;0FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QAs  
GH,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAAsB,IAAI,CAtGa,SAsgZ,CAAU,OAAV,CAAL,C;YAAyB,WAAy,WAA  
I,OAAJ,C;;QAtG3D,OAuGO,W;O;KA7GX,C;0FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QAUgh,Q;  
QAAhB,iD;UAAgB,cAAhB,e;UAAAsB,IAAI,CAvGW,SAuGV,CAAU,OAAV,CAAL,C;YAAyB,WAAy,WAAI,O  
AAJ,C;;QAvG3D,OAwGO,W;O;KA9GX,C;0FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QAwGH,Q;Q  
AAhB,iD;UAAgB,cAAhB,e;UAAAsB,IAAI,CAxGY,SAwGX,CAAU,OAAV,CAAL,C;YAAyB,WAAy,WAAI,OA  
AJ,C;;QAxG3D,OAyGO,W;O;KA/GX,C;0FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QAYGH,Q;QAA  
hB,iD;UAAgB,cAAhB,e;UAAAsB,IAAI,CAzGa,SAyGZ,CAAU,OAAV,CAAL,C;YAAyB,WAAy,WAAI,OAAJ,C;  
;QAzG3D,OA0GO,W;O;KAhHX,C;0FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QA0GH,Q;QAAhB,i  
D;UAAgB,cAAhB,e;UAAAsB,IAAI,CA1Gc,SA0Gb,CAAU,OAAV,CAAL,C;YAAyB,WAAy,WAAI,OAAJ,C;;QA  
1G3D,OA2GO,W;O;KAjHX,C;0FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QA2GH,Q;QAAhB,iD;UA  
AgB,cAAhB,e;UAAAsB,IAAI,CA3Ge,SA2Gd,CAAU,OAAV,CAAL,C;YAAyB,WAAy,WAAI,OAAJ,C;;QA3G3D,  
OA4GO,W;O;KAiHX,C;0FASA,yB;MAAA,+D;MA4GA,oC;MAAA,gC;MA5GA,uC;QAMW,kBAAY,gB;QA4G  
H,Q;QAAhB,iD;UAAgB,cAAhB,0B;UAAAsB,IAAI,CA5GY,SA4GX,CAAU,oBAAV,CAAL,C;YAAyB,WAAy,W  
AAI,oBAAJ,C;;QA5G3D,OA6GO,W;O;KAnHX,C;IASA,kC;MAMI,OAAO,2BAAGB,gBAAhB,C;K;IAGX,iD;M  
AMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,eAAJ,C;UAAqB,WAAy,WAA  
I,OAAJ,C;;MACvD,OAAO,W;K;4FAGX,6C;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,  
M;QAAsB,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAy,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;8FAGX,  
6C;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,OAAV,CA  
AL,C;UAAyB,WAAy,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;8FAGX,6C;MAMoB,Q;MAAhB,wBAAGB,SAAhB  
,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAy,WAAI,OAAJ,C;;M  
AC3D,OAAO,W;K;8FAGX,6C;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IA  
AI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAy,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;8FAGX,6C;MAMoB,  
Q;MAAhB,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAy  
B,WAAy,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;8FAGX,6C;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAgB,  
cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAy,WAAI,OAAJ,C;;MAC3D,OAA  
O,W;K;8FAGX,6C;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,CAAC,U  
AAU,OAAV,CAAL,C;UAAyB,WAAy,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;8FAGX,6C;MAMoB,Q;MAAhB,  
wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAy,  
WAAI,OAAJ,C;;MAC3D,OAAO,W;K;8FAGX,yB;MAAA,oC;MAAA,gC;MAAA,oD;QAMoB,Q;QAAhB,wBAA  
gB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UAAAsB,IAAI,CAAC,UAAU,oBAAV,CAAL,C;YAAyB,WAA  
Y,WAAI,oBAAJ,C;;QAC3D,OAAO,W;O;KAPX,C;sFAUA,6C;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAg  
B,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAy,WAAI,OAAJ,C;;MAC1D,OAAO,W;  
K;wFAGX,6C;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAA  
V,CAAJ,C;UAAwB,WAAy,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;wFAGX,6C;MAMoB,Q;MAAhB,wBAAGB,S  
AAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAy,WAAI,OAAJ,C;;M  
AC1D,OAAO,W;K;wFAGX,6C;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IA  
AI,UAAU,OAAV,CAAJ,C;UAAwB,WAAy,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;wFAGX,6C;MAMoB,Q;MA  
AhB,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAy,W  
AAI,OAAJ,C;;MAC1D,OAAO,W;K;wFAGX,6C;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAgB,cAAA,SA

hB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;wFAGX,6C  
;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UA  
AwB,WAAY,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;wFAGX,6C;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QA  
AgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAI,OAAJ,C;;MAC1D,OAAO,  
W;K;wFAGX,yB;MAAA,oC;MAAA,gC;MAAA,oD;QAMoB,Q;QAaHb,wBAAgB,SAAhB,gB;UAAgB,cAAhB,U  
AAgB,SAAhB,O;UAAAsB,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,WAAY,WAAI,oBAAJ,C;;QAC1D,OAAO,W;  
O;KAPX,C;IAUA,mC;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OKvtIe,W;;MLwtItC,OAA4D,OAArD,yBAAY,OAA  
Q,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,CAAqD,C;K;IAGhE,qC;MAIL,IAAI,OAAQ,UAAZ,C;  
QAAuB,OK/tIe,W;;MLguItC,OgBpsIsC,OhBosI/B,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAA  
vB,IAA3B,CgBpsI+B,C;K;IhBusI1C,qC;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OKvule,W;;MLwuItC,OgBpsIuC,O  
hBosIhC,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,CgBpsIgC,C;K;IhBusI3C,qC;  
MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OK/uIe,W;;MLgvItC,OgBpsIqC,OhBosI9B,yBAAY,OAAQ,MAApB,EAA2  
B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,CgBpsI8B,C;K;IhBusIzC,qC;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OKv  
vIe,W;;MLwvItC,OgBpsIsC,OhBosI/B,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,  
CgBpsI+B,C;K;IhBusI1C,qC;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OK/vIe,W;;MLgwItC,OgBpsIuC,OhBosIhC,y  
BAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,CgBpsIgC,C;K;IhBusI3C,qC;MAIL,IAAI,  
OAAQ,UAAZ,C;QAAuB,OKvwIe,W;;MLwwItC,OgBpsIwC,OhBosIjC,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,  
aAAR,GAAuB,CAAvB,IAA3B,CgBpsIiC,C;K;IhBusI5C,qC;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OK/wIe,W;;M  
LgxItC,OgBpsIyC,OhBosIlC,0BAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,CgBpsIkC,  
C;K;IhBusI7C,qC;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OKvxIe,W;;MLwxItC,OAA4D,SAArD,0BAAY,OAAQ,  
MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,CAAqD,C;K;IAGhE,qC;MAOkB,Q;MAHd,WAAmB,w  
BAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAae,OAAO,W;MACTb,WAAW,iBAaA,IAA  
b,C;MACG,yB;MAAd,OAAc,cAAAd,C;QAac,uB;QACV,IAAK,WAAI,UAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;I  
AGX,qC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAae,  
OAAO,W;MACTb,WAAW,iBAAgB,IAAhB,C;MACG,yB;MAAd,OAAc,cAAAd,C;QAac,uB;QACV,IAAK,WAAI,  
UAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAx  
B,C;MACnB,IAAI,SAAQ,CAAZ,C;QAae,OAAO,W;MACTb,WAAW,iBAAiB,IAAjB,C;MACG,yB;MAAd,OAA  
c,cAAAd,C;QAac,uB;QACV,IAAK,WAAI,UAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAOkB,Q;MAHd,  
WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAae,OAAO,W;MACTb,WAAW,i  
BAaE,IAAf,C;MACG,yB;MAAd,OAAc,cAAAd,C;QAac,uB;QACV,IAAK,WAAI,UAAI,KAAJ,CAAJ,C;;MAET,O  
AAO,I;K;IAGX,sC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ  
,C;QAae,OAAO,W;MACTb,WAAW,iBAAgB,IAAhB,C;MACG,yB;MAAd,OAAc,cAAAd,C;QAac,uB;QACV,IA  
AK,WAAI,UAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EA  
AwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAae,OAAO,W;MACTb,WAAW,iBAAiB,IAAjB,C;MACG,yB;  
MAAd,OAAc,cAAAd,C;QAac,uB;QACV,IAAK,WAAI,UAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAO  
kB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAae,OAAO,W;MA  
CTb,WAAW,iBAAkB,IAAiB,C;MACG,yB;MAAd,OAAc,cAAAd,C;QAac,uB;QACV,IAAK,WAAI,UAAI,KAAJ,C  
AAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,I  
AAI,SAAQ,CAAZ,C;QAae,OAAO,W;MACTb,WAAW,iBAAmB,IAAnB,C;MACG,yB;MAAd,OAAc,cAAAd,C;Q  
AAc,uB;QACV,IAAK,WAAI,UAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAOkB,Q;MAHd,WAAmB,w  
BAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAae,OAAO,W;MACTb,WAAW,iBAAgB,IA  
AhB,C;MACG,yB;MAAd,OAAc,cAAAd,C;QAac,uB;QACV,IAAK,WAAI,sBAAI,KAAJ,EAAJ,C;;MAET,OAAO,I  
;K;IAGX,wC;MAMwB,UACT,M;MAHX,aAAa,aAAa,SAAb,EAAmB,OAAQ,KAA3B,C;MACb,kBAAkB,C;MA  
CE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHb,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK,WAA  
L,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,cAAU,OAAQ,KAAIB,C;MACb,kBAAkB,C  
;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHb,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK,WAA  
AL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,eAAW,OAAQ,KAAIB,C;MACb,kBAA  
kB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHb,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK

,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,eAAS,OAAQ,KAAjB,C;MACb,kB  
AAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UA  
AK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAU,OAAQ,KAAIB,C;MAC  
b,kBAAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAAP,YAAw  
B,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAW,OAAQ,KAAiB,  
C;MACb,kBAAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAAP,  
YAAwB,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAZ,OAAQ,KA  
ApB,C;MACb,kBAAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4B  
AAP,YAAwB,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,oBAAa,OAA  
Q,KAArB,C;MACb,kBAAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAA  
O,4BAAP,YAAwB,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAU,  
OAAQ,KAAIB,C;MACb,kBAAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,  
EAAO,4BAAP,YAAwB,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,  
OAAO,yBAAZ,CAAZ,EAAe,CAAf,C;MAC9B,OAAO,yBAAZ,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,  
CAAvB,IAA3B,C;K;IAGX,0C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,cAAU,CAAV,C;MAC9B,OAAO,yB  
AAZ,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,  
C;QAAuB,OAAO,eAAW,CAAX,C;MAC9B,OAAO,yBAAZ,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CA  
AvB,IAA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,eAAS,CAAT,C;MAC9B,OAAO,yBAA  
Z,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;Q  
AAuB,OAAO,iBAAU,CAAV,C;MAC9B,OAAO,yBAAZ,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAv  
B,IAA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,iBAAW,CAAX,C;MAC9B,OAAO,yBAAZ  
,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QA  
AuB,OAAO,iBAAZ,CAAZ,C;MAC9B,OAAO,yBAAZ,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IA  
AA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,oBAAa,CAAb,C;MAC9B,OAAO,0BAAZ,OA  
AQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB  
,OAAO,iBAAU,CAAV,C;MAC9B,OAAO,0BAAZ,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA  
3B,C;K;IAGX,4B;MACiB,Q;MeloJb,IAAI,Ef4nJI,KAAK,Ce5nJT,CAAJ,C;QACI,cf2nJc,sD;Qe1nJd,MAAM,gCA  
AyB,OAAQ,WAAjC,C;;Mf2nJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAA  
e,OAAO,iB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;M  
ACZ,WAAW,iBAAa,CAAb,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAL,IAAJ,C;Q  
ACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MACiB,Q;MexpJb,IAAI,EfkpJI,KAAK,Celp  
JT,CAAJ,C;QACI,cfipJc,sD;QehpJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MfipJV,IAAI,MAAK,CAAT,C;QAAY,  
OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OA  
AO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAgB,CAAhB,C;MACX,wBAAa,SAAb,gB;Q  
AAa,WAAA,SAAb,M;QACI,IAAK,WAAL,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;I  
AGX,8B;MACiB,Q;Me9qJb,IAAI,EfwqJI,KAAK,CexqJT,CAAJ,C;QACI,cfuqJc,sD;QetqJd,MAAM,gCAAYB,OA  
AQ,WAAjC,C;;MfuqJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAAe,OAAO,  
mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WA  
AW,iBAAiB,CAAjB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAL,IAAJ,C;QACL,IA  
AI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MACiB,Q;MepsJb,IAAI,Ef8rJI,KAAK,Ce9rJT,CAAJ  
,C;QACI,cf6rJc,sD;Qe5rJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mf6rJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W  
;MACnB,IAAI,KAAK,gBAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAA  
K,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAe,CAAf,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,S  
AAb,M;QACI,IAAK,WAAL,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MACi  
B,Q;Me1tJb,IAAI,EfotJI,KAAK,CeptJT,CAAJ,C;QACI,cfmtJc,sD;QeltJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mf  
mtJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAAe,OAAO,mB;MACtB,IAAI,  
MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAgB,CA  
AhB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAL,IAAJ,C;QACL,IAAI,mCAAW,CA



Af,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;MehvJb,IAAI,Ef0uJI,KAAK,Ce1uJT,CAAJ,C;QACI,cfyuJc,sD;QexuJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MfyuJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAAe,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAiB,CAAjB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;MetwJb,IAAI,EfgwJI,KAAK,CehwJT,CAAJ,C;QACI,cf+vJc,sD;Qe9vJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mf+vJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAAe,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAkB,CAAIB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;Me5xJb,IAAI,EfsxJI,KAAK,CetxJT,CAAJ,C;QACI,cfqxJc,sD;QepxJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MfqxJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAAe,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAmB,CAAnB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;MelzJb,IAAI,Ef4yJI,KAAK,Ce5yJT,CAAJ,C;QACI,cf2yJc,sD;Qe1yJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mf2yJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAAe,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,CAAL,EAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAgB,CAAhB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAb,UAAa,SAAb,O;QACI,IAAK,WAAI,iBAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,gC;Me1zJI,IAAI,Efk0JI,KAAK,Cel0JT,CAAJ,C;QACI,cfi0Jc,sD;Qeh0Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mfi0JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,iB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAaA,CAAb,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Me70JI,IAAI,Efq1JI,KAAK,Cer1JT,CAAJ,C;QACI,cf01Jc,sD;Qen1Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mfo1JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAgB,CAAhB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Meh2JI,IAAI,Efw2JI,KAAK,Cex2JT,CAAJ,C;QACI,cfu2Jc,sD;Qet2Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mfu2JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAiB,CAAjB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Men3JI,IAAI,Ef23JI,KAAK,Ce33JT,CAAJ,C;QACI,cf03Jc,sD;Qez3Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mf03JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBA Ae,CAAf,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Met4JI,IAAI,Ef84JI,KAAK,Ce94JT,CAAJ,C;QACI,cf64Jc,sD;Qe54Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mf64JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAgB,CAAhB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Mez5JI,IAAI,Efi6JI,KAAK,Cej6JT,CAAJ,C;QACI,cf6Jc,sD;Qe/5Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mfg6JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAiB,CAAjB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Me56JI,IAAI,Efo7JI,KAAK,Cep7JT,CAAJ,C;QACI,cfm7Jc,sD;Qel7Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mfm7JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAkB,CAAIB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WA

AI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Me/7JI,IAAI,Efu8JI,KAAK,Cev8JT,CAAJ,C;QACI,cfs8J  
c,sD;Qer8Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mfs8JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WA  
AW,gB;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,U  
AAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAmB,CAAnB,C;MACX,iBAAc,OAAO,CAAP,IAAd,U  
AA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;MeI9JI,IAAI,Ef09JI,K  
AAK,CeI9JT,CAAJ,C;QACI,cfy9Jc,sD;Qex9Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mfy9JV,IAAI,MAAK,CAA  
T,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACTb,IAAI,MAA  
K,CAAT,C;QAAY,OAAO,OAAO,sBAAK,OAAO,CAAP,IAAL,EAAP,C;MACnB,WAAW,iBAAgB,CAAhB,C;M  
ACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,sBAAK,KAAL,EAAP,C;MACT,OAAO,  
I;K;gGAGX,yB;MAAA,8D;MAAA,4C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,I  
AAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO  
,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,2C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/  
B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;Q  
AGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WA  
A+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,  
IAAL,C;;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,gD;MAAA,uC;QAMI,iBAAc,  
wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,Q  
AAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,gD;MAAA,uC;Q  
AMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAA  
O,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,gD;  
MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C  
;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;  
MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CA  
AV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8  
D;MAAA,4C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAA  
K,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,y  
B;MAAA,8D;MAAA,oC;MAAA,4C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IA  
AI,CAAC,UAAU,sBAAK,KAAL,EAAP,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,i  
B;O;KAXX,C;wFAcA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAaA,SAAb,gB;UAAa,  
WAAA,SAAb,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO  
,I;O;KAZX,C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAaA,SAAb,gB;UAAa,  
WAAA,SAAb,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO  
,I;O;KAZX,C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAaA,SAAb,gB;UAAa,  
WAAA,SAAb,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO  
,I;O;KAZX,C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAaA,SAAb,gB;UAAa,  
WAAA,SAAb,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO  
,I;O;KAZX,C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAaA,SAAb,gB;UAAa,  
WAAA,SAAb,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO  
,I;O;KAZX,C;0FAeA,yB;MAAA,+D;MAAA,oC;MAAA,gC;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wB  
AAa,SAAb,gB;UAAa,WAAb,UAAa,SAAb,O;UACI,IAAI,CAAC,UAAU,iBAAV,CAAL,C;YACI,K;UACJ,IAAK,  
WAAI,iBAAJ,C;;QAET,OAAO,I;O;KAZX,C;IAeA,4B;MAII,eAAe,CAAC,mBAAO,CAAP,IAAD,IAAa,CAAb,I;  
MACf,IAAI,WAAW,CAAf,C;QAAkB,M;MACIB,mBAAmB,wB;MACnB,iBAAc,CAAd,WAAiB,QAAjB,U;QAC



B,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAA  
L,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,mD;MAWI,oCAAA,2BAaKB,SAA  
IB,EAA6B,OAA7B,EAAc,gBAAtC,C;MACb,eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,  
cAAa,QAAjB,C;QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QAC  
I,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,  
mC;;K;IAIR,6B;MAII,IA+nEO,qBAAQ,CA+nEf,C;QAAe,OAAO,W;MACtB,WAAW,wB;MACN,WAAL,IAAK,C  
;MACL,OAAO,I;K;IAGX,+B;MAII,IA6nEO,qBAAQ,CA7nEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,W  
AAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IA2nEO,qBAAQ,CA3nEf,C;QAAe,OAAO,W;MACtB,WAAW,  
0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAynEO,qBAAQ,CAznEf,C;QAAe,OAAO,W;M  
ACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAunEO,qBAAQ,CAvnEf,C;QAA  
e,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAqnEO,qBAAQ,C  
ArnEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAmn  
EO,qBAAQ,CAnnEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,  
+B;MAII,IAinEO,qBAAQ,CAjnEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAA  
O,I;K;IAGX,+B;MAII,IA+mEO,qBAAQ,CA/mEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,  
C;MACL,OAAO,I;K;IAGX,kC;MAII,IAqiEO,qBAAQ,CariEf,C;QAAe,OAAO,S;MACtB,aAAa,aAAa,SAAb,EA  
AmB,gBAAnB,C;MACb,gBAAgB,wB;MACHb,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAA  
P,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IAiiEO,qBAAQ,CAjiiEf,C;QAAe,OAAO,S;MA  
CtB,aAAa,cAAU,gBAAV,C;MACb,gBAAgB,0B;MACHb,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,C  
AAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IA6hEO,qBAAQ,CA7hEf,C;QAAe,O  
AAO,S;MACtB,aAAa,eAAW,gBAAX,C;MACb,gBAAgB,0B;MACHb,aAAU,CAAV,OAAa,SAAb,M;QACI,OAA  
O,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IAyhEO,qBAAQ,CAzhEf,  
C;QAAe,OAAO,S;MACtB,aAAa,eAAS,gBAAT,C;MACb,gBAAgB,0B;MACHb,aAAU,CAAV,OAAa,SAAb,M;Q  
ACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IAqhEO,qBAA  
Q,CArhEf,C;QAAe,OAAO,S;MACtB,aAAa,iBAAU,gBAAV,C;MACb,gBAAgB,0B;MACHb,aAAU,CAAV,OAA  
a,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IAi  
hEO,qBAAQ,CAjhEf,C;QAAe,OAAO,S;MACtB,aAAa,iBAAW,gBAAX,C;MACb,gBAAgB,0B;MACHb,aAAU,C  
AAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,o  
C;MAII,IA6gEO,qBAAQ,CA7gEf,C;QAAe,OAAO,S;MACtB,aAAa,iBAAy,gBAAZ,C;MACb,gBAAgB,0B;MAC  
hB,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,  
M;K;IAGX,oC;MAII,IAygEO,qBAAQ,CAzgEf,C;QAAe,OAAO,S;MACtB,aAAa,oBAAa,gBAAb,C;MACb,gBAA  
gB,0B;MACHb,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC  
5B,OAAO,M;K;IAGX,oC;MAII,IAqgEO,qBAAQ,CArgEf,C;QAAe,OAAO,S;MACtB,aAAa,iBAAU,gBAAV,C;M  
ACb,gBAAgB,0B;MACHb,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CA  
AL,C;MAC5B,OAAO,M;K;IAGX,4B;MAKI,qBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,qBAAQ,4BAAR,C;K;IAGJ,8  
B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,  
8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAG  
J,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,sC;MAOI,aAAU,wBAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBA  
AQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,U  
AAK,CAAL,IAAU,I;;K;IAIIB,sC;MAOI,aAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,  
CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,  
IAAU,I;;K;IAIIB,uC;MAOI,aAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IA  
AR,C;QACf,WAAW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU  
I;;K;IAIIB,uC;MAOI,aAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QA  
Cf,WAAW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIIB,  
uC;MAOI,aAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAA  
W,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIIB,uC;MA  
OI,aAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,

CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIIB,uC;MAOI,aAAU,  
0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,C;  
QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIIB,uC;MAOI,aAAU,0BAAV,  
OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,C;QACX,U  
AAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIIB,uC;MAOI,aAAU,0BAAV,OAA2B,C  
AA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,C;QACX,UAAK,CA  
AL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;kFAlIB,yB;MAAA,oD;MiB15LA,sC;MAAA,oC;M  
AAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,C  
AAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjBm5Lf,sC;QAMI,IAAI,mBAAO,CAAX,C;UAAc,oBi  
Bz5Ld,eAAW,iBjBy5LsB,QiBz5LtB,CAAX,CjBy5Lc,C;;O;KANIB,C;sGASA,yB;MAAA,oD;MiBh5LA,sC;MAA  
A,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/  
EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjBy4Lf,sC;QAMI,IAAI,mBAAO,CAAX,C;UAA  
c,oBiB/4Ld,eAAW,2BjB+4LgC,QiB/4LhC,CAAX,CjB+4Lc,C;;O;KANIB,C;IASA,mC;MAMI,oBAAS,cAAT,C;K;  
IAGJ,qC;MAII,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;K;IAIR,qC;MAII,IAAI,mBAAO,CAAX,C;QACI,e;Q  
ACA,oB;;K;IAIR,qC;MAII,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;K;IAIR,qC;MAII,IAAI,mBAAO,CAAX,  
C;QACI,iB;QACA,oB;;K;IAIR,qC;MAII,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;K;IAIR,qC;MAII,IAAI,mB  
AAO,CAAX,C;QACI,e;QACA,oB;;K;IAIR,qC;MAII,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;K;IAIR,2B;MA  
MI,OAAqB,OAAd,sBAAc,C;K;IAGzB,6B;MAI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAuC,OkBnhMhC,WIBm  
hMgC,C;K;IAG3C,6B;MAI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAuC,OkB1hMhC,WIB0hMgC,C;K;IAG3C,6  
B;MAI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAuC,OkBjiMhC,WIBiiMgC,C;K;IAG3C,6B;MAI0B,kBAAf,yB;M  
AAuB,mB;MAA9B,OAAuC,OkBxiMhC,WIBwiMgC,C;K;IAG3C,6B;MAI0B,kBAAf,yB;MAAuB,mB;MAA9B,O  
AAuC,OkB/iMhC,WIB+iMgC,C;K;IAG3C,6B;MAI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAuC,OkBtjMhC,WIB  
sjMgC,C;K;IAG3C,6B;MAI0B,kBAAf,0B;MAAuB,mB;MAA9B,OAAuC,OkB7jMhC,WIB6jMgC,C;K;IAG3C,gC  
;MAMI,IA6kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MACD,kBAAd,SgB3jKiB,Q;MhB2jKK,mB;MAA7B,OkBv  
kMO,W;K;IIB0kMX,kC;MAII,IA6kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MACD,kBAAd,SgBzjKiB,Q;MhByj  
KK,iB;MAA7B,OkB/kMO,W;K;IIBkIMX,kC;MAII,IA6kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MACD,kBAAd,  
SgBvjKiB,Q;MhBujKK,iB;MAA7B,OkBvIMO,W;K;IIB0IMX,kC;MAII,IA6kDO,qBAAQ,CA7kDf,C;QAAe,OAA  
O,S;MACD,kBAAd,SgBrjKiB,Q;MhBqjKK,iB;MAA7B,OkB/IMO,W;K;IIBkmMX,kC;MAII,IA6kDO,qBAAQ,C  
A7kDf,C;QAAe,OAAO,S;MACD,kBAAT,UAAL,SAAK,C;MAAiB,mB;MAA7B,OkBvmMO,W;K;IIB0mMX,kC;  
MAII,IA6kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MACD,kBAAd,SgBljKiB,Q;MhBkjKK,iB;MAA7B,OkB/mM  
O,W;K;IIBknMX,kC;MAII,IA6kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MACD,kBAAd,SgBhjKiB,Q;MhBgjKK,  
iB;MAA7B,OkBvnMO,W;K;IIB0nMX,kC;MAII,IAqIDO,qBAAQ,CARIdf,C;QAAe,OAAO,S;MACD,kBAAT,UA  
AL,SAAK,C;MAAiB,iB;MAA7B,OkB/nMO,W;K;IIBkoMX,0C;MAMI,IA2gDO,qBAAQ,CA3gDf,C;QAAe,OAA  
O,S;MACD,kBAAd,SgB7nKiB,Q;MhB6nKK,sBAAS,cAAT,C;MAA7B,OkBzoMO,W;K;IIB4oMX,4C;MAII,IA2g  
DO,qBAAQ,CA3gDf,C;QAAe,OAAO,S;MACD,kBAAd,SgB3nKiB,Q;MhB2nKK,6B;MAA7B,OkBjpMO,W;K;IIB  
BopMX,4C;MAII,IA2gDO,qBAAQ,CA3gDf,C;QAAe,OAAO,S;MACD,kBAAd,SgBznKiB,Q;MhBjnKK,6B;MA  
A7B,OkBzpMO,W;K;IIB4pMX,4C;MAII,IA2gDO,qBAAQ,CA3gDf,C;QAAe,OAAO,S;MACD,kBAAd,SgBvnKi  
B,Q;MhBunKK,6B;MAA7B,OkBjqMO,W;K;IIBoqMX,4C;MAII,IA2gDO,qBAAQ,CA3gDf,C;QAAe,OAAO,S;M  
ACD,kBAAT,UAAL,SAAK,C;MAAiB,6B;MAA7B,OkBzqMO,W;K;IIB4qMX,4C;MAII,IA2gDO,qBAAQ,CA3g  
Df,C;QAAe,OAAO,S;MACD,kBAAd,SgBpnKiB,Q;MhBonKK,6B;MAA7B,OkBjrMO,W;K;IIBorMX,4C;MAII,I  
A2gDO,qBAAQ,CA3gDf,C;QAAe,OAAO,S;MACD,kBAAd,SgBlnKiB,Q;MhBknKK,6B;MAA7B,OkBzrMO,W;  
K;IIB4rMX,4C;MAII,IAmhDO,qBAAQ,CAnhdF,C;QAAe,OAAO,S;MACD,kBAAT,UAAL,SAAK,C;MAAiB,6B;  
MAA7B,OkBjsMO,W;K;IIBosMX,gD;MAMI,IAy8CO,qBAAQ,CAz8Cf,C;QAAe,OAAO,S;MACD,kBAAd,SgB/r  
KiB,Q;MhB+rKK,iC;MAA7B,OkB3sMO,W;K;sFIB8sMX,yB;MAAA,wD;MiBnsMA,sC;MAAA,oC;MAAA,uBA  
Oe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA  
2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjB4rMf,sC;QAQI,OAAO,sBiBpsMP,eAAW,iBjBosMiB,QiBpsM  
jB,CAAX,CjBosMO,C;O;KARX,C;wFAWA,yB;MAAA,wD;MiB9sMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,  
8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,C

A5DN,CAA3B,C;W;S;OA4DI,C;MjBusMf,sC;QAMI,OAAO,sBiB7sMP,eAAW,iBjB6sMiB,QiB7sMjB,CAAX,CjB6sMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MiBvtMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjBgtMf,sC;QAMI,OAAO,sBiBttMP,eAAW,iBjBstMiB,QiBttMjB,CAAX,CjBstMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MiBhuMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjBytMf,sC;QAMI,OAAO,sBiB/tMP,eAAW,iBjB+tMiB,QiB/tMjB,CAAX,CjB+tMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MiBzuMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjBkuMf,sC;QAMI,OAAO,sBiBxuMP,eAAW,iBjBwuMiB,QiBxuMjB,CAAX,CjBwuMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MiBlvMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjB2uMf,sC;QAMI,OAAO,sBiBjvMP,eAAW,iBjBivMiB,QiBjvMjB,CAAX,CjBivMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MiB3vMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjBovMf,sC;QAMI,OAAO,sBiB1vMP,eAAW,iBjB0vMiB,QiB1vMjB,CAAX,CjB0vMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MiBpwMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjB6vMf,sC;QAMI,OAAO,sBiBnwMP,eAAW,iBjBmwMiB,QiBnwMjB,CAAX,CjBmwMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MiB7wMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjBswMf,sC;QAMI,OAAO,sBiB5wMP,eAAW,iBjB4wMiB,QiB5wMjB,CAAX,CjB4wMO,C;O;KANX,C;0GASA,yB;MAAA,wD;MiBnwMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjB4vMf,sC;QAMI,OAAO,sBiBlwMP,eAAW,2BjBkwM2B,QiBlwM3B,CAAX,CjBkwMO,C;O;KANX,C;4GASA,yB;MAAA,wD;MiB5wMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjBqwMf,sC;QAI,OAAO,sBiBzwMP,eAAW,2BjBywM2B,QiBzwM3B,CAAX,CjBywMO,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MiBnxMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjB4wMf,sC;QAI,OAAO,sBiBhxMP,eAAW,2BjBgxM2B,QiBhxM3B,CAAX,CjBgxMO,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MiB1xMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjBmxMf,sC;QAI,OAAO,sBiBvxMP,eAAW,2BjBuxM2B,QiBvxM3B,CAAX,CjBuxMO,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MiBjyMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjB0xmMf,sC;QAI,OAAO,sBiB9xMP,eAAW,2BjB8xM2B,QiB9xM3B,CAAX,CjB8xMO,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MiBxyMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjBiyMf,sC;QAI,OAAO,sBiBryMP,eAAW,2BjBqyM2B,QiBryM3B,CAAX,CjBqyMO,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MiB/yMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjBwyMf,sC;QAI,OAAO,sBiB5yMP,eAAW,2BjB4yM2B,QiB5yM3B,CAAX,CjB4yMO,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MiBtzMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjB+yMf,sC;QAI,OAAO,sBiBnzMP,eAAW,2BjBmzM2B,QiBnzM3B,CAAX,CjBmzMO,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MiB7zMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjBszMf,sC;QAI,OAAO,sBiB1zMP,eAAW,2BjB0zM2B,QiB1zM3B,CAAAX,CjB0zMO,C;O;KAJX,C;IAOA,qC;MAMI,OAAO,sBAAW,cAAAX,C;K;IAGX,uC;MAIoB,kBgBz1KQ,iB;MhBy

1KA,iB;MAAxB,OAAiC,WkB/2M1B,WIB+2M0B,C;K;IAGrC,uC;MAIoB,kBgBt1KQ,iB;MhBs1KA,iB;MAAxB,  
OAAiC,WkBt3M1B,WIBs3M0B,C;K;IAGrC,uC;MAIoB,kBgBn1KQ,iB;MhBm1KA,iB;MAAxB,OAAiC,WkB73  
M1B,WIB63M0B,C;K;IAGrC,uC;MAIoB,kBAAT,oB;MAAiB,mB;MAAxB,OAAiC,WkBp4M1B,WIBo4M0B,C;  
K;IAGrC,uC;MAIoB,kBgB90KQ,iB;MhB80KA,iB;MAAxB,OAAiC,WkB34M1B,WIB24M0B,C;K;IAGrC,uC;M  
AIoB,kBgB30KQ,iB;MhB20KA,iB;MAAxB,OAAiC,WkB15M1B,WIBk5M0B,C;K;IAGrC,uC;MAIoB,kBAAT,oB  
;MAAiB,iB;MAAxB,OAAiC,WkBz5M1B,WIBy5M0B,C;K;IAGrC,2C;MAMI,OAAmC,OAA5B,2BAAgB,UAAh  
B,CAA4B,C;K;IAGvC,6C;MAIoB,kBAAf,yB;MAAuB,iC;MAA9B,OAAqD,OkBz6M9C,WIBy6M8C,C;K;IAGzD  
,6C;MAIoB,kBAAf,yB;MAAuB,iC;MAA9B,OAAqD,OkBh7M9C,WIBg7M8C,C;K;IAGzD,6C;MAIoB,kBAAf,yB  
;MAAuB,iC;MAA9B,OAAqD,OkBv7M9C,WIBu7M8C,C;K;IAGzD,6C;MAIoB,kBAAf,yB;MAAuB,iC;MAA9B,  
OAAqD,OkB97M9C,WIB87M8C,C;K;IAGzD,6C;MAIoB,kBAAf,yB;MAAuB,iC;MAA9B,OAAqD,OkBr8M9C,  
WIBq8M8C,C;K;IAGzD,6C;MAIoB,kBAAf,yB;MAAuB,iC;MAA9B,OAAqD,OkB58M9C,WIB48M8C,C;K;IAGz  
D,6C;MAIoB,kBAAf,yB;MAAuB,iC;MAA9B,OAAqD,OkBn9M9C,WIBm9M8C,C;K;IAGzD,6C;MAIoB,kBAAf,  
oB;MAAuB,iC;MAA9B,OAAqD,OkB19M9C,WIB09M8C,C;K;IAkoCrD,gC;MAAQ,oBAAS,CAAT,EAAY,wBA  
AZ,C;K;IAMR,kC;MAAQ,oBAAS,CAAT,EAAY,0BAAZ,C;K;IAMR,kC;MAAQ,oBAAS,CAAT,EAAY,0BAAZ,  
C;K;IAMR,kC;MAAQ,oBAAS,CAAT,EAAY,0BAAZ,C;K;IAMR,kC;MAAQ,oBAAS,CAAT,EAAY,0BAAZ,C;K;  
IAMR,kC;MAAQ,oBAAS,CAAT,EAAY,0BAAZ,C;K;IAMR,kC;MAAQ,oBAAS,CAAT,EAAY,0BAAZ,C;K;IA  
MR,kC;MAAQ,oBAAS,CAAT,EAAY,0BAAZ,C;K;IAMR,kC;MAAQ,oBAAS,CAAT,EAAY,0BAAZ,C;K;oFAE  
Z,qB;MAKI,OAAO,qBAAQ,C;K;sFAGnB,qB;MAKI,OAAO,qBAAQ,C;K;sFAGnB,qB;MAKI,OAAO,qBAAQ,C;  
K;sFAGnB,qB;MAKI,OAAO,qBAAQ,C;K;sFAGnB,qB;MAKI,OAAO,qBAAQ,C;K;sFAGnB,qB;MAKI,OAAO,q  
BAAQ,C;K;sFAGnB,qB;MAKI,OAAO,qBAAQ,C;K;sFAGnB,qB;MAKI,OAAO,qBAAQ,C;K;sFAGnB,qB;MAKI  
,OAAO,qBAAQ,C;K;oFAGnB,qB;MAKI,OAAO,EAxEa,qBAAQ,CAwER,C;K;4FAGX,qB;MAKI,OAAO,EAxE  
A,qBAAQ,CAwER,C;K;4FAGX,qB;MAKI,OAAO,EAxEa,qBAAQ,CAwER,C;K;4FAGX,qB;MAKI,OAAO,EA  
EA,qBAAQ,CAwER,C;K;4FAGX,qB;MAKI,OAAO,EAxEa,qBAAQ,CAwER,C;K;4FAGX,qB;MAKI,OAAO,EA  
xEa,qBAAQ,CAwER,C;K;4FAGX,qB;MAKI,OAAO,EAxEa,qBAAQ,CAwER,C;K;4FAGX,qB;MAKI,OAAO,E  
AxEA,qBAAQ,CAwER,C;K;4FAGX,qB;MAKI,OAAO,EAxEa,qBAAQ,CAwER,C;K;IAOP,kC;MAAQ,0BAAO,  
CAAP,I;K;IAMR,oC;MAAQ,0BAAO,CAAP,I;K;IAMR,oC;MAAQ,0BAAO,CAAP,I;K;IAMR,oC;MAAQ,0BAA  
O,CAAP,I;K;IAMR,oC;MAAQ,0BAAO,CAAP,I;K;IAMR,oC;MAAQ,0BAAO,CAAP,I;K;IAMR,oC;MAAQ,0BAA  
AO,CAAP,I;K;IAMR,oC;MAAQ,0BAAO,CAAP,I;K;IAMR,oC;MAAQ,0BAAO,CAAP,I;K;IA8TZ,yD;MAcI,sBA  
AS,cAAT,EAAYB,SAAzB,EAAoC,OAApC,C;K;IAGJ,yD;MAYI,mBAAK,SAAL,EAAGB,OAAhB,C;MACA,qB  
AAQ,SAAR,EAAMB,OAAhB,C;K;IAGJ,yD;MAYI,mBAAK,SAAL,EAAGB,OAAhB,C;MACA,sBAAQ,SAAR,EAAMB,  
OAAhB,C;K;IAGJ,0D;MAYI,mBAAK,SAAL,EAAGB,OAAhB,C;MACA,sBAAQ,SAAR,EAAMB,OAAhB,C;K;IAGJ,0  
D;MAYI,mBAAK,SAAL,EAAGB,OAAhB,C;MACA,sBAAQ,SAAR,EAAMB,OAAhB,C;K;IAGJ,0D;MAYI,mBA  
AK,SAAL,EAAGB,OAAhB,C;MACA,sBAAQ,SAAR,EAAMB,OAAhB,C;K;IAGJ,0D;MAYI,mBAAK,SAAL,EA  
AGB,OAAhB,C;MACA,sBAAQ,SAAR,EAAMB,OAAhB,C;K;IA2B0B,oD;MAAA,wB;QAAW,2BAAK,KAAL,C;  
O;K;IAJzC,mC;MAII,OAAO,qBAAa,gBAAb,EAAMB,gCAAnB,C;K;IAOgB,8C;MAAA,wB;QAAW,wBAAK,K  
AAL,C;O;K;IAJtC,gC;MAII,OAAO,+BAAU,gBAAV,GAAgB,6BAAhB,C;K;IAOgB,8C;MAAA,wB;QAAW,wB  
AAK,KAAL,C;O;K;IAJtC,gC;MAII,OAAO,kBAAU,gBAAV,EAAGB,6BAAhB,C;K;IAOkB,kD;MAAA,wB;QAA  
W,0BAAK,KAAL,C;O;K;IAJxkC,kC;MAII,OAAO,kCAAY,gBAAZ,GAAkB,+BAAIB,C;K;IAOiB,gD;MAAA,wB;  
QAAW,yBAAK,KAAL,C;O;K;IAJvC,iC;MAII,OAAO,kCAAW,gBAAZ,GAAiB,8BAAjB,C;K;IAOe,4C;MAAA,  
wB;QAAW,uBAAK,KAAL,C;O;K;IAJrC,+B;MAII,OAAO,gCAAS,gBAAT,GAAe,4BAAf,C;K;IAOgB,8C;MAA  
A,wB;QAAW,wBAAK,KAAL,C;O;K;IAJtC,gC;MAII,OAAO,kBAAU,gBAAV,EAAGB,6BAAhB,C;K;IAOiB,gD;  
MAAA,wB;QAAW,yBAAK,KAAL,C;O;K;IAJvC,iC;MAII,OAAO,gCAAW,gBAAZ,GAAiB,8BAAjB,C;K;wFA2  
CX,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAAY,gBAAZ,CAAKB,EAAC,EAAd  
,C;QAC1B,kBAAY,mBAAoB,QAAPB,C;QAyqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,  
CAAU,OAAV,C;UM7+QnB,wBAAL,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;QNm0PA,OA4qBO,W;O;K;AXrBX,  
C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAAY,gBAAZ,CAAKB,EAAC,  
EAAd,C;QAC1B,kBAAY,mBAAoB,QAAPB,C;QAyqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0

qB/B,CAAU,OAAV,C;UM5/QnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;QNk1PA,OA4qBO,W;O;KAxrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAKB,EAAC,EAAD,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,CAAU,OAAV,C;UM3gRnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;QNi2PA,OA4qBO,W;O;KAxrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAKB,EAAC,EAAD,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,CAAU,OAAV,C;UM1hRnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;QNg3PA,OA4qBO,W;O;KAxrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAKB,EAAC,EAAD,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,CAAU,OAAV,C;UMziRnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;QN+3PA,OA4qBO,W;O;KAxrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAKB,EAAC,EAAD,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,CAAU,OAAV,C;UMxjRnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;QN84PA,OA4qBO,W;O;KAxrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAKB,EAAC,EAAD,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,CAAU,OAAV,C;UMvkRnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;QN65PA,OA4qBO,W;O;KAxrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAKB,EAAC,EAAD,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,CAAU,OAAV,C;UMtlRnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;QN46PA,OA4qBO,W;O;KAxrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;MA4qBA,oC;MAAA,gC;MA5qBA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAKB,EAAC,EAAD,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,0B;UACI,WA1qB8C,SA0qB/B,CAAU,oBAAV,C;UMrmRnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;QN27PA,OA4qBO,W;O;KAxrBX,C;4FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAKB,EAAC,EAAD,C;QAC1B,kBAAC,mBAAoB,QAApB,C;QAmQL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAy,aApQoC,WAOqHC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QApQhB,OAsQO,W;O;KAIRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAKB,EAAC,EAAD,C;QAC1B,kBAAC,mBAAuB,QAAvB,C;QAOQL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAy,aArQuC,WaqQnC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QArQhB,OAuQO,W;O;KANRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAKB,EAAC,EAAD,C;QAC1B,kBAAC,mBAAwB,QAAxB,C;QAqQL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAy,aAtQwC,WAsQpC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QAtQhB,OAuQO,W;O;KAPRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAKB,EAAC,EAAD,C;QAC1B,kBAAC,mBAAyB,QAAzB,C;QAYQL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAy,aA1QyC,WA0QrC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QA1QhB,OA4QO,W;O;KAxRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAKB,EAAC,EAAD,C;QAC1B,kBAAC,mBAA0B,QAA1B,C;QA0QL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAy,aA3Q0C,WA2QtC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QA3QhB,OA6QO,W;O;KAZRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MA6QA,oC;MAAA,gC;MA7QA,yC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAKB,EAAC,EAAD,C;QAC1B,kBAAC,mBAAuB,QAAvB,C;QA2QL,Q;QAAhB,iD;UAAgB,cAAhB,0B;UACI,WAAy,aA5QuC,WA4QnC,CAAY,oBAAZ,CAAJ,EAA0B,oBAA1B,C;;QA5QhB,OA8QO,W;O;KA1RX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MA







AhB,M;QACI,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;sGAGX,iD;MAWoB,Q;M  
AAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;  
MAEhB,OAAO,W;K;sGAGX,iD;MAWoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,W  
AAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;sGAGX,iD;MAWoB,Q;MAAhB,wBAAgB,  
SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,  
W;K;sGAGX,iD;MAWoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aAAI,OAA  
J,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;sGAGX,iD;MAWoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAA  
gB,cAAA,SAAhB,M;QACI,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;sGAGX,iD;  
MAWoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aAAI,OAAJ,EAAa,cAAc,O  
AAd,CAAb,C;;MAEhB,OAAO,W;K;sGAGX,yB;MAAA,oC;MAAA,gC;MAAA,wD;QAWoB,Q;QAAhB,wBAAg  
B,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,WAAY,aAAI,oBAAJ,EAAa,cAAc,oBAAd,CAAb,C;;QA  
EhB,OAAO,W;O;KADx,C;IAiBA,8C;MAIiB,Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAAY  
,WAAI,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QA  
CI,WAAY,WAAI,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAA,S  
AAb,M;QACI,WAAY,WAAI,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,Q;MAAb,wBAAa,SAAb,gB;QAA  
a,WAAA,SAAb,M;QACI,WAAY,WAAI,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,Q;MAAb,wBAAa,SA  
Ab,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,WAAI,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,Q;MAAb,w  
BAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,WAAI,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,  
Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,WAAI,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,  
gD;MAIiB,Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,WAAI,IAAJ,C;;MAEhB,OAAO,  
W;K;IAGX,gD;MAIiB,Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAb,UAAa,SAAb,O;QACI,WAAY,WAAI,iBAAJ,C  
;;MAEhB,OAAO,W;K;IAGX,8B;MAII,OAAO,wBAAa,eAAW,YAAY,gBAAZ,CAAX,CAAb,C;K;IAGX,gC;MAI  
I,OAAO,0BAAa,eAAc,YAAY,gBAAZ,CAAd,CAAb,C;K;IAGX,gC;MAII,OAAO,0BAAa,eAAe,YAAY,gBAAZ,  
CAAf,CAAb,C;K;IAGX,gC;MAII,OAAO,0BAAa,eAAa,YAAY,gBAAZ,CAAb,CAAb,C;K;IAGX,gC;MAII,OAA  
O,0BAAa,eAAc,YAAY,gBAAZ,CAAd,CAAb,C;K;IAGX,gC;MAII,OAAO,0BAAa,eAAe,YAAY,gBAAZ,CAAf,  
CAAb,C;K;IAGX,gC;MAII,OAAO,0BAAa,eAAgB,YAAY,gBAAZ,CAAhB,CAAb,C;K;IAGX,gC;MAII,OAAO,0  
BAAa,eAAiB,YAAY,gBAAZ,CAAjB,CAAb,C;K;IAGX,gC;MAII,OAAO,0BAAa,eAAc,YAAiB,eAAL,gBAAK,E  
AAa,GAAb,CAAjB,CAAd,CAAb,C;K;IAGX,2B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;  
UAAL,K;aACA,C;UAAK,cAAO,UAAK,CAAL,CAAP,C;UAAL,K;;UACa,qBAAL,SAAK,C;UAHV,K;;MAAP,W  
;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,UA  
AK,CAAL,CAAP,C;UAAL,K;;UACa,uBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,  
QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,UAAK,CAAL,CAAP,C;UAAL,K;;UACa  
,uBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,k  
B;UAAL,K;aACA,C;UAAK,cAAO,UAAK,CAAL,CAAP,C;UAAL,K;;UACa,uBAAL,SAAK,C;UAHV,K;;MAAP,  
W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,U  
AAK,CAAL,CAAP,C;UAAL,K;;UACa,uBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,6B;MAIiB,IAAN,I;MAA  
A,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,UAAK,CAAL,CAAP,C;UAAL,K;;UA  
Ca,uBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK  
,kB;UAAL,K;aACA,C;UAAK,cAAO,UAAK,CAAL,CAAP,C;UAAL,K;;UACa,uBAAL,SAAK,C;UAHV,K;;MAA  
P,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,  
UAAK,CAAL,CAAP,C;UAAL,K;;UACa,uBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,6B;MAIiB,IAAN,I;MA  
AA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,sBAAK,CAAL,EAAP,C;UAAL,K;;  
UACa,uBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,kC;MAII,OAAO,iBA Ae,aAAL,SAAK,CAAf,C;K;IAGX,o  
C;MAKiB,Q;MADb,WAAW,iBAAgB,gBAAhB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAMb,I  
AAK,WAAI,IAAJ,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iBAAiB,gBAAjB,C;MACX,wBA  
Aa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAMb,IAAK,WAAI,IAAJ,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;  
MADb,WAAW,iBA Ae,gBAAf,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAMb,IAAK,WAAI,IAAJ  
,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iBAAgB,gBAAhB,C;MACX,wBAAa,SAAb,gB;QA

Aa,WAAA,SAAb,M;QAAMb,IAAK,WAAI,IAAJ,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iB  
AAiB,gBAAjB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAMb,IAAK,WAAI,IAAJ,C;;MACxB,O  
AAO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iBAAkB,gBAaIB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SA  
Ab,M;QAAMb,IAAK,WAAI,IAAJ,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iBAAMb,gBAA  
nB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAMb,IAAK,WAAI,IAAJ,C;;MACxB,OAAO,I;K;IA  
GX,oC;MAKiB,Q;MADb,WAAW,iBAAgB,gBAAhB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAb,UAAa,SAAb,O;  
QAAMb,IAAK,WAAI,iBAAJ,C;;MACxB,OAAO,I;K;IAGX,0B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aAC  
H,C;UAAK,iB;UAAL,K;aACA,C;UAAK,aAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,+BAAa,qBAAiB,YA  
AY,gBAAZ,CAAjB,CAAb,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH  
,C;UAAK,iB;UAAL,K;aACA,C;UAAK,aAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBAAoB,YAA  
Y,gBAAZ,CAApB,CAAb,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,  
C;UAAK,iB;UAAL,K;aACA,C;UAAK,aAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBAAqB,YAAY  
,gBAAZ,CAArB,CAAb,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;  
UAAK,iB;UAAL,K;aACA,C;UAAK,aAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBAAMb,YAAY,  
gBAAZ,CAAnB,CAAb,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;  
UAAK,iB;UAAL,K;aACA,C;UAAK,aAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBAAoB,YAAY,g  
BAAZ,CAApB,CAAb,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;U  
AAK,iB;UAAL,K;aACA,C;UAAK,aAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBAAqB,YAAY,gB  
AAZ,CAArB,CAAb,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UA  
AK,iB;UAAL,K;aACA,C;UAAK,aAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBAAsB,YAAY,gBA  
AZ,CAAtB,CAAb,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAA  
K,iB;UAAL,K;aACA,C;UAAK,aAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBAAuB,YAAY,gBAA  
Z,CAAvB,CAAb,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK  
,iB;UAAL,K;aACA,C;UAAK,aAM,sBAAK,CAAL,EAAN,C;UAAL,K;;UACQ,iCAAa,qBAAoB,YAAiB,eAAL,  
gBAAK,EAAa,GAAb,CAAjB,CAApB,CAAb,C;UAHL,K;;MAAP,W;K;oFAOJ,yB;MAAA,+D;MAwaA,gD;MAx  
aA,uC;QAMW,kBAAU,gB;QAsaD,Q;QAaHb,iD;UAAgB,cAAhB,e;UACI,WAv6B,SAualB,CAAU,OAAV,C;U  
ACC,OAAZ,WAAy,EAAO,IAAP,C;;QAxahB,OA0aO,W;O;KAhbX,C;sFASA,yB;MAAA,+D;MA0aA,gD;MA1a  
A,uC;QAMW,kBAAU,gB;QAwaD,Q;QAaHb,iD;UAAgB,cAAhB,e;UACI,WAZ6B,SAyalB,CAAU,OAAV,C;UA  
CC,OAAZ,WAAy,EAAO,IAAP,C;;QA1ahB,OA4aO,W;O;KAlbX,C;sFASA,yB;MAAA,+D;MA4aA,gD;MA5aA,  
uC;QAMW,kBAAU,gB;QA0aD,Q;QAaHb,iD;UAAgB,cAAhB,e;UACI,WA3a6B,SA2alB,CAAU,OAAV,C;UAC  
C,OAAZ,WAAy,EAAO,IAAP,C;;QA5ahB,OA8aO,W;O;KApbX,C;sFASA,yB;MAAA,+D;MA8aA,gD;MA9aA,u  
C;QAMW,kBAAU,gB;QA4aD,Q;QAaHb,iD;UAAgB,cAAhB,e;UACI,WA7a6B,SA6alB,CAAU,OAAV,C;UACC,  
OAAZ,WAAy,EAAO,IAAP,C;;QA9ahB,OAgbO,W;O;KAtbX,C;sFASA,yB;MAAA,+D;MAgbA,gD;MAhbA,uC;  
QAMW,kBAAU,gB;QA8aD,Q;QAaHb,iD;UAAgB,cAAhB,e;UACI,WA/a6B,SA+a1B,CAAU,OAAV,C;UACC,O  
AAZ,WAAy,EAAO,IAAP,C;;QAhhbB,OAkbO,W;O;KAxbX,C;sFASA,yB;MAAA,+D;MAkbA,gD;MA1bA,uC;Q  
AMW,kBAAU,gB;QAgbD,Q;QAaHb,iD;UAAgB,cAAhB,e;UACI,Wajb6B,SAib1B,CAAU,OAAV,C;UACC,OA  
AZ,WAAy,EAAO,IAAP,C;;QA1bhB,OAobO,W;O;KA1bX,C;sFASA,yB;MAAA,+D;MAobA,gD;MApbA,uC;QA  
MW,kBAAU,gB;QAkbD,Q;QAaHb,iD;UAAgB,cAAhB,e;UACI,WAnb6B,SAmb1B,CAAU,OAAV,C;UACC,OA  
AZ,WAAy,EAAO,IAAP,C;;QAphbB,OAsbO,W;O;KA5bX,C;sFASA,yB;MAAA,+D;MASbA,gD;MATbA,uC;QA  
MW,kBAAU,gB;QAobD,Q;QAaHb,iD;UAAgB,cAAhB,e;UACI,WArb6B,SAq1B,CAAU,OAAV,C;UACC,OAA  
Z,WAAy,EAAO,IAAP,C;;QAtbhB,OAwbO,W;O;KA9bX,C;sFASA,yB;MAAA,+D;MAwbA,oC;MAAA,gD;MA  
AA,gC;MAxBa,uC;QAMW,kBAAU,gB;QAsbD,Q;QAaHb,iD;UAAgB,cAAhB,0B;UACI,WAvb6B,SAub1B,CAA  
U,oBAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAxhbB,OA0bO,W;O;KAhcX,C;sFASA,yB;MAAA,+D;MA  
0bA,gD;MA1bA,uC;QAUW,kBAAU,gB;QAwbD,Q;QAaHb,iD;UAAgB,cAAhB,e;UACI,WAZb6B,SAyb1B,CAA  
U,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QA1bhB,OA4bO,W;O;KAtcX,C;kGAaA,yB;MAAA,+D;MAS  
JA,gD;MATJA,uC;QAYW,kBAAiB,gB;QAqJR,gB;QADhB,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WAtJoC  
,SAsJzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAmB,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAvJhB,OAY  
JO,W;O;KArKX,C;oGAeA,yB;MAAA,+D;MAyJA,gD;MAZJA,uC;QAYW,kBAAiB,gB;QAwJR,gB;QADhB,YAA

Y,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WaZJoC,SAyJzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAhB,C;UA  
CC,OAAZ,WAAy,EAAO,IAAP,C;;QA1JhB,OA4JO,W;O;KAxKX,C;oGAeA,yB;MAAA,+D;MA4JA,gD;MA5JA,  
uC;QAYW,kBAaiB,gB;QA2JR,gB;QADhB,YAAy,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WA5JoC,SA4JzB,EAA  
U,cAAV,EAAU,sBAAV,WAAmB,OAAhB,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QA7JhB,OA+JO,W;O;KA3  
KX,C;oGAeA,yB;MAAA,+D;MA+JA,gD;MA/JA,uC;QAYW,kBAaiB,gB;QA8JR,gB;QADhB,YAAy,C;QACZ,i  
D;UAAgB,cAAhB,e;UACI,WA/JoC,SA+JzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAhB,C;UACC,OAAZ,W  
AAy,EAAO,IAAP,C;;QAhKhB,OAKKO,W;O;KA9KX,C;oGAeA,yB;MAAA,+D;MAkKA,gD;MAiKA,uC;QAY  
W,kBAaiB,gB;QAiKR,gB;QADhB,YAAy,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WAlKoC,SAkKzB,EAAU,cAA  
V,EAAU,sBAAV,WAAmB,OAAhB,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAnKhB,OAqKO,W;O;KAjLX,C;  
oGAeA,yB;MAAA,+D;MAqKA,gD;MArKA,uC;QAYW,kBAaiB,gB;QAoKR,gB;QADhB,YAAy,C;QACZ,iD;U  
AAgB,cAAhB,e;UACI,WArKoC,SAqKzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAhB,C;UACC,OAAZ,WAA  
y,EAAO,IAAP,C;;QAtKhB,OAwKO,W;O;KAplX,C;oGAeA,yB;MAAA,+D;MAwKA,gD;MAxKA,uC;QAYW,  
kBAaiB,gB;QAuKR,gB;QADhB,YAAy,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WAxKoC,SAwKzB,EAAU,cAAV,  
EAAU,sBAAV,WAAmB,OAAhB,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAzKhB,OA2KO,W;O;KAvLX,C;o  
GAeA,yB;MAAA,+D;MA2KA,gD;MA3KA,uC;QAYW,kBAaiB,gB;QA0KR,gB;QADhB,YAAy,C;QACZ,iD;UA  
AgB,cAAhB,e;UACI,WA3KoC,SA2KzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAhB,C;UACC,OAAZ,WAA  
y,EAAO,IAAP,C;;QA5KhB,OA8KO,W;O;KA1LX,C;oGAeA,yB;MAAA,+D;MA8KA,oC;MAAA,gD;MAAA,gC;  
MA9KA,uC;QAYW,kBAaiB,gB;QA6KR,gB;QADhB,YAAy,C;QACZ,iD;UAAgB,cAAhB,OB;UACI,WA9KoC,S  
A8KzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,oBAAnB,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QA/KhB,OAI  
LO,W;O;KA7LX,C;oGAeA,yB;MAAA,+D;MAiLA,gD;MAjLA,uC;QAYW,kBAaiB,gB;QAglR,gB;QADhB,YA  
AY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WAjLoC,SAiLzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAhB,C;U  
ACC,OAAZ,WAAy,EAAO,IAAP,C;;QAILhB,OAoLO,W;O;KAhMX,C;sGAeA,yB;MAAA,gD;MAAA,oD;QAW  
oB,UACS,M;QAFzB,YAAy,C;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,WAAU,c  
AAV,EAAU,sBAAV,WAAmB,OAAhB,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;u  
GakBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAy,C;QACZ,wBAAgB,SAAhB,gB;UAAgB,c  
AAA,SAAhB,M;UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAhB,C;UACC,OAAZ,WAAy,EAAO,  
IAAP,C;;QAEhB,OAAO,W;O;KafX,C;wGakBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAy,  
C;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAm  
B,OAAhB,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;wGakBA,yB;MAAA,gD;MA  
AA,oD;QAWoB,UACS,M;QAFzB,YAAy,C;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAA  
W,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAhB,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;  
O;KafX,C;wGakBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAy,C;QACZ,wBAAgB,SAAhB,g  
B;UAAgB,cAAA,SAAhB,M;UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAhB,C;UACC,OAAZ,W  
AAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;wGakBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QA  
FzB,YAAy,C;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,WAAU,cAAV,EAAU,sBA  
AV,WAAmB,OAAhB,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;wGakBA,yB;MA  
AA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAy,C;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;  
UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAhB,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEh  
B,OAAO,W;O;KafX,C;wGakBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAy,C;QACZ,wBAA  
gB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAhB,C;UA  
CC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;wGakBA,yB;MAAA,oC;MAAA,gD;MAAA,g  
C;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAy,C;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAh  
B,O;UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,oBAAnB,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;  
QAEhB,OAAO,W;O;KafX,C;wGakBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAy,C;QACZ,  
wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAhB  
C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;uFAkBA,yB;MAAA,gD;MAAA,oD;QAI  
oB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,UAAU,OAaV,C;UACC,OAAZ,W  
AAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;OFaWA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAhB,wBA



ACL,IAAI,aAAJ,C;YACH,aNiiVuC,gB;YAA5B,WMhiVX,aNgiVgC,GMhiVhC,EAAS,MAAT,C;YACA,e;;YAE A,c;;UN6hVA,iB;UACA,IAAK,WAAI,oBAAJ,C;;QA5QT,OA8QO,W;O;KAvRX,C;sFAyA,yB;MAAA,wE;MA8 QA,+D;MA9QA,yD;QAUW,kBAAU,oB;QA8QD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UA/QiD,WA+QvC,CA AY,OAAZ,C;UMnjVP,U;UADP,YNqjVe,WMrjVH,WNqjVwB,GMrjVxB,C;UACL,IAAI,aAAJ,C;YACH,aNmjVu C,gB;YAA5B,WMIjVX,aNkjVgC,GMljVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN+iVA,iB;UACA,IAAK,Waj RyD,cAiRrD,CAAe,OAAf,CAAJ,C;;QAJRT,OAmRO,W;O;KA7RX,C;sFAaA,yB;MAAA,wE;MAmRA,+D;MANR A,yD;QAUW,kBAAU,oB;QAmRD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UApRiD,WAOvC,CAAY,OAAZ,C;U MrkVP,U;UADP,YNukVe,WMvkVH,WNukVwB,GMvkVxB,C;UACL,IAAI,aAAJ,C;YACH,aNqkVuC,gB;YAA5 B,WMpkVX,aNokVgC,GMpkVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNikVA,iB;UACA,IAAK,WAtRyD,cAs RrD,CAAe,OAAf,CAAJ,C;;QAtRT,OAwRO,W;O;KAISX,C;uFAaA,yB;MAAA,wE;MAwRA,+D;MAxRA,yD;QA UW,kBAAU,oB;QAwRD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UAzRiD,WAYvC,CAAY,OAAZ,C;UMvIVP,U ;UADP,YNyIVe,WMzIVH,WNyIVwB,GMzIVxB,C;UACL,IAAI,aAAJ,C;YACH,aNulVuC,gB;YAA5B,WMtlVX, aNslVgC,GMtlVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNmlVA,iB;UACA,IAAK,WA3RyD,cA2RrD,CAAe,O Aaf,CAAJ,C;;QA3RT,OA6RO,W;O;KAvSX,C;uFAaA,yB;MAAA,wE;MA6RA,+D;MA7RA,yD;QAUW,kBAAU, oB;QA6RD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UA9RiD,WA8RvC,CAAY,OAAZ,C;UMzmVP,U;UADP,YN2 mVe,WM3mVH,WN2mVwB,GM3mVxB,C;UACL,IAAI,aAAJ,C;YACH,aNymVuC,gB;YAA5B,WMxmVX,aNw mVgC,GMxmVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNqmVA,iB;UACA,IAAK,WAhSyD,cAgSrD,CAAe,OA Af,CAAJ,C;;QAhST,OAKSO,W;O;KA5SX,C;uFAaA,yB;MAAA,wE;MAkSA,+D;MAISA,yD;QAUW,kBAAU,oB ;QAKSD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UAnSiD,WAmSvC,CAAY,OAAZ,C;UM3nVP,U;UADP,YN6nV e,WM7nVH,WN6nVwB,GM7nVxB,C;UACL,IAAI,aAAJ,C;YACH,aN2nVuC,gB;YAA5B,WM1nVX,aN0nVgC,G M1nVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNunVA,iB;UACA,IAAK,WArSyD,cAqSrD,CAAe,OAAf,CAAJ, C;;QArST,OAuSO,W;O;KAjTX,C;uFAaA,yB;MAAA,wE;MAuSA,+D;MAvSA,yD;QAUW,kBAAU,oB;QAU SD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UAXSiD,WAWsvC,CAAY,OAAZ,C;UM7oVP,U;UADP,YN+oVe,WM/oV H,WN+oVwB,GM/oVxB,C;UACL,IAAI,aAAJ,C;YACH,aN6oVuC,gB;YAA5B,WM5oVX,aN4oVgC,GM5oVhC, EAAS,MAAT,C;YACA,e;;YAEA,c;;UNyoVA,iB;UACA,IAAK,WA1SyD,cA0SrD,CAAe,OAAf,CAAJ,C;;QA1ST ,OA4SO,W;O;KAiTX,C;uFAaA,yB;MAAA,wE;MA4SA,+D;MA5SA,yD;QAUW,kBAAU,oB;QA4SD,Q;QAAhB, iD;UAAgB,cAAhB,e;UACI,UA7SiD,WA6SvC,CAAY,OAAZ,C;UM/pVP,U;UADP,YNiqVe,WMjqVH,WNiqVw B,GMjqVxB,C;UACL,IAAI,aAAJ,C;YACH,aN+pVuC,gB;YAA5B,WM9pVX,aN8pVgC,GM9pVhC,EAAS,MAA T,C;YACA,e;;YAEA,c;;UN2pVA,iB;UACA,IAAK,WA/SyD,cA+SrD,CAAe,OAAf,CAAJ,C;;QA/ST,OAiTO,W;O ;KA3TX,C;uFAaA,yB;MAAA,wE;MAiTA,+D;MAjTA,yD;QAUW,kBAAU,oB;QAItd,Q;QAAhB,iD;UAAgB,cA AhB,e;UACI,UAItd,WAKtvC,CAAY,OAAZ,C;UMjrVP,U;UADP,YNmrVe,WMnrVH,WNmrVwB,GMnrVxB, C;UACL,IAAI,aAAJ,C;YACH,aNirVuC,gB;YAA5B,WMhrVX,aNgrVgC,GMhrVhC,EAAS,MAAT,C;YACA,e;;Y AEA,c;;UN6qVA,iB;UACA,IAAK,WApTyD,cAoTrD,CAAe,OAAf,CAAJ,C;;QApTT,OAsTO,W;O;KAhUX,C;uF AaA,yB;MAAA,wE;MAStA,oC;MAAA,+D;MAAA,gC;MAiTA,yD;QAUW,kBAAU,oB;QAStd,Q;QAAhB,iD;U AAgB,cAAhB,OB;UACI,UAvTiD,WAutvC,CAAY,oBAAZ,C;UMnsVP,U;UADP,YNqsVe,WMrsVH,WNqsVwB, GMrsVxB,C;UACL,IAAI,aAAJ,C;YACH,aNmsVuC,gB;YAA5B,WMIsvX,aNksVgC,GMIsVhC,EAAS,MAAT,C; YACA,e;;YAEA,c;;UN+rVA,iB;UACA,IAAK,WAZtyD,cAyTrD,CAAe,oBAAf,CAAJ,C;;QAZTT,OA2TO,W;O;K ArUX,C;wFAaA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M; UACI,UAAU,YAAY,OAAZ,C;UMz5UP,U;UADP,YN25Ue,WM35UH,WN25UwB,GM35UxB,C;UACL,IAAI,aA AJ,C;YACH,aNy5UuC,gB;YAA5B,WMx5UX,aNw5UgC,GMx5UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNq5 UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q ;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UM16UP,U;UADP,YN 46Ue,WM56UH,WN46UwB,GM56UxB,C;UACL,IAAI,aAAJ,C;YACH,aN06UuC,gB;YAA5B,WMz6UX,aNy6Ug C,GMz6UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNs6UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W; O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB, M;UACI,UAAU,YAAY,OAAZ,C;UM37UP,U;UADP,YN67Ue,WM77UH,WN67UwB,GM77UxB,C;UACL,IAAI, aAAJ,C;YACH,aN27UuC,gB;YAA5B,WM17UX,aN07UgC,GM17UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNu 7UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,

Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAAY,OAAZ,C;UM58UP,U;UADP,Y N88Ue,WM98UH,WN88UwB,GM98UxB,C;UACL,IAAI,aAAJ,C;YACH,aN48UuC,gB;YAA5B,WM38UX,aN28 UgC,GM38UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNw8UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO, W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAAY,OAAZ,C;UM79UP,U;UADP,YN+9Ue,WM/9UH,WN+9UwB,GM/9UxB,C;UACL,IA AI,aAAJ,C;YACH,aN69UuC,gB;YAA5B,WM59UX,aN49UgC,GM59UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U Ny9UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QAS oB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAAY,OAAZ,C;UM9+UP,U;UAD P,YNg/Ue,WMh/UH,WNg/UwB,GMh/UxB,C;UACL,IAAI,aAAJ,C;YACH,aN8+UuC,gB;YAA5B,WM7+UX,aN6 +UgC,GM7+UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN0+UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO ,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAAY,OAAZ,C;UM//UP,U;UADP,YNigVe,WMjgVH,WNigVwB,GMjgVxB,C;UACL,IAAI, aAAJ,C;YACH,aN+/UuC,gB;YAA5B,WM9/UX,aN8/UgC,GM9/UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN2/U A,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q; QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAAY,OAAZ,C;UMhhVP,U;UADP,YNk hVe,WMlhVH,WNkhVwB,GMlhVxB,C;UACL,IAAI,aAAJ,C;YACH,aNghVuC,gB;YAA5B,WM/gVX,aN+gVgC, GM/gVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN4gVA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O; KAdX,C;0FAiBA,yB;MAAA,oC;MAAA,+D;MAAA,gC;MAAA,sD;QASoB,Q;QAAhB,wBAAgB,SAAhB,gB;UA AgB,cAAhB,UAAgB,SAAhB,O;UACI,UAAU,YAAAY,oBAAZ,C;UMjiVP,U;UADP,YNmiVe,WMniVH,WNmiV wB,GMniVxB,C;UACL,IAAI,aAAJ,C;YACH,aNiiVuC,gB;YAA5B,WMhiVX,aNgiVgC,GMhiVhC,EAAS,MAAT ,C;YACA,e;;YAEA,c;;UN6hVA,iB;UACA,IAAK,WAAI,oBAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;M AAA,+D;MAAA,sE;QAUoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAAY,O AAZ,C;UMnjVP,U;UADP,YNqjVe,WMrjVH,WNqjVwB,GMrjVxB,C;UACL,IAAI,aAAJ,C;YACH,aNmjVuC,gB; YAA5B,WMIjVX,aNkjVgC,GMljVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN+iVA,iB;UACA,IAAK,WAAI,eA Ae,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;0FAkBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAAhB,wBAAg B,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAAY,OAAZ,C;UMrkVP,U;UADP,YNukVe,WMvkVH, WNUkVwB,GMvkVxB,C;UACL,IAAI,aAAJ,C;YACH,aNqkVuC,gB;YAA5B,WMpkVX,aNokVgC,GMpkVhC,E AAS,MAAT,C;YACA,e;;YAEA,c;;UNikVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;K AfX,C;2FAkBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M; UACI,UAAU,YAAAY,OAAZ,C;UMvIVP,U;UADP,YNyIVe,WMzIVH,WNyIVwB,GMzIVxB,C;UACL,IAAI,aAAJ, C;YACH,aNulVuC,gB;YAA5B,WMtlVX,aNslVgC,GMtlVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNmlVA,iB;U ACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;2FAkBA,yB;MAAA,+D;MAAA,sE;QAU oB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAAY,OAAZ,C;UMzmVP,U;UAD P,YN2mVe,WM3mVH,WN2mVwB,GM3mVxB,C;UACL,IAAI,aAAJ,C;YACH,aNymVuC,gB;YAA5B,WMxmV X,aNwmVgC,GMxmVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNqmVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,C AAJ,C;;QAET,OAAO,W;O;KafX,C;2FAkBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAAhB,wBAAgB,SAAhB,g B;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAAY,OAAZ,C;UM3nVP,U;UADP,YN6nVe,WM7nVH,WN6nVwB, GM7nVxB,C;UACL,IAAI,aAAJ,C;YACH,aN2nVuC,gB;YAA5B,WM1nVX,aN0nVgC,GM1nVhC,EAAS,MAAT, C;YACA,e;;YAEA,c;;UNunVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;2FAk BA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU, YAAAY,OAAZ,C;UM7oVP,U;UADP,YN+oVe,WM/oVH,WN+oVwB,GM/oVxB,C;UACL,IAAI,aAAJ,C;YACH,a N6oVuC,gB;YAA5B,WM5oVX,aN4oVgC,GM5oVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNyoVA,iB;UACA,I AAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;2FAkBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;Q AAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAAY,OAAZ,C;UM/pVP,U;UADP,YNiqV e,WMjqVH,WNiqVwB,GMjqVxB,C;UACL,IAAI,aAAJ,C;YACH,aN+pVuC,gB;YAA5B,WM9pVX,aN8pVgC,G M9pVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN2pVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,O AAO,W;O;KafX,C;2FAkBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA ,SAAhB,M;UACI,UAAU,YAAAY,OAAZ,C;UMjrVP,U;UADP,YNmrVe,WMnrVH,WNmrVwB,GMnrVxB,C;UAC



L,IAAI,aAAJ,C;YACH,aNirVuC,gB;YAA5B,WMhrVX,aNgrVgC,GMhrVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;  
;UN6qVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;2FakBA,yB;MAAA,oC;M  
AAA,+D;MAAA,gC;MAAA,sE;QAUoB,Q;QAAbB,wBAAGB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;U  
ACI,UAAU,YAAY,oBAAZ,C;UMnsVP,U;UADP,YNqsVe,WMrsVH,WNqsVwB,GMrsVxB,C;UACL,IAAI,aAAJ  
,C;YACH,aNmsVuC,gB;YAA5B,WMIsvX,aNksVgC,GMIsvhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN+rVA,iB  
;UACA,IAAK,WAAI,eAAe,oBAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;0FakBA,yB;MAAA,kC;MAAA,4C;M  
AAA,wE;QAQW,sC;QAAA,8C;O;MARX,oDASQ,Y;QAA6C,OAAgB,qBAAhB,oBAAgB,C;O;MATrE,iDAUQ,  
mB;QAAoC,gCAAY,OAAZ,C;O;MAV5C,gF;MAAA,yC;QAQI,2D;O;KARJ,C;4EAca,yB;MAAA,gE;MAAA,uC;  
QAOW,kBAAM,eAAa,gBAAb,C;QA+UA,Q;QAAb,iD;UAAa,WAAb,e;UACI,WAAy,WAhViB,SAGVb,CAAU,I  
AAV,CAAJ,C;;QAhVhB,OaiVO,W;O;KaxVX,C;8EAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,gBA  
Ab,C;QA+UA,Q;QAAb,iD;UAAa,WAAb,e;UACI,WAAy,WAhViB,SAGVb,CAAU,IAAV,CAAJ,C;;QAhVhB,OA  
iVO,W;O;KaxVX,C;8EAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,gBAAb,C;QA+UA,Q;QAAb,iD;  
UAAa,WAAb,e;UACI,WAAy,WAhViB,SAGVb,CAAU,IAAV,CAAJ,C;;QAhVhB,OaiVO,W;O;KaxVX,C;8EAU  
A,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,gBAAb,C;QA+UA,Q;QAAb,iD;UAAa,WAAb,e;UACI,WAA  
Y,WAhViB,SAGVb,CAAU,IAAV,CAAJ,C;;QAhVhB,OaiVO,W;O;KaxVX,C;8EAUA,yB;MAAA,gE;MAAA,uC  
;QAOW,kBAAM,eAAa,gBAAb,C;QA+UA,Q;QAAb,iD;UAAa,WAAb,e;UACI,WAAy,WAhViB,SAGVb,CAAU,I  
AAV,CAAJ,C;;QAhVhB,OaiVO,W;O;KaxVX,C;8EAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,gBA  
Ab,C;QA+UA,Q;QAAb,iD;UAAa,WAAb,e;UACI,WAAy,WAhViB,SAGVb,CAAU,IAAV,CAAJ,C;;QAhVhB,OA  
iVO,W;O;KaxVX,C;8EAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,gBAAb,C;QA+UA,Q;QAAb,iD;  
UAAa,WAAb,e;UACI,WAAy,WAhViB,SAGVb,CAAU,IAAV,CAAJ,C;;QAhVhB,OaiVO,W;O;KaxVX,C;8EAU  
A,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,gBAAb,C;QA+UA,Q;QAAb,iD;UAAa,WAAb,e;UACI,WAA  
Y,WAhViB,SAGVb,CAAU,IAAV,CAAJ,C;;QAhVhB,OaiVO,W;O;KaxVX,C;8EAUA,yB;MAAA,gE;MAiVA,o  
C;MAAA,gC;MAjVA,uC;QAOW,kBAAM,eAAa,gBAAb,C;QA+UA,Q;QAAb,iD;UAAa,WAAb,0B;UACI,WAA  
Y,WAhViB,SAGVb,CAAU,iBAAV,CAAJ,C;;QAhVhB,OaiVO,W;O;KaxVX,C;0FAUA,yB;MAAA,gE;MAAA,u  
C;QAOW,kBAaA,eAAa,gBAAb,C;QAgHP,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UACI,WAAy,WAjH  
wB,SAiHpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QAJHhB,OAkHO,W;O;KAZHX,C;4FAUA,  
yB;MAAA,gE;MAAA,uC;QAOW,kBAaA,eAAa,gBAAb,C;QAmHP,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb  
,e;UACI,WAAy,WApHwB,SAoHpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QApHhB,OAqHO  
,W;O;KA5HX,C;4FAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAaA,eAAa,gBAAb,C;QAsHP,gB;QADb,YAAY,C;  
QACZ,iD;UAAa,WAAb,e;UACI,WAAy,WAvHwB,SAuHpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CA  
AJ,C;;QAvHhB,OAwhO,W;O;KA/HX,C;4FAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAaA,eAAa,gBAAb,C;QA  
yHP,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UACI,WAAy,WA1HwB,SA0HpB,EAAU,cAAV,EAAU,sBA  
AV,WAAmB,IAAnB,CAAJ,C;;QA1HhB,OA2HO,W;O;KAIIX,C;4FAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBA  
Aa,eAAa,gBAAb,C;QA4HP,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UACI,WAAy,WA7HwB,SA6HpB,E  
AAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QA7HhB,OA8HO,W;O;KARIX,C;2FAUA,yB;MAAA,gE;  
MAAA,uC;QAOW,kBAaA,eAAa,gBAAb,C;QA+HP,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UACI,WAA  
Y,WAhIwB,SAGIpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QAhIhB,OaiIO,W;O;KaxIX,C;4F  
AUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAaA,eAAa,gBAAb,C;QakIP,gB;QADb,YAAY,C;QACZ,iD;UAAa,W  
AAb,e;UACI,WAAy,WAnIwB,SAmIpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QAnIhB,OAoi  
O,W;O;KA3IX,C;4FAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAaA,eAAa,gBAAb,C;QAqIP,gB;QADb,YAAY,C  
;QACZ,iD;UAAa,WAAb,e;UACI,WAAy,WAtIwB,SAsIpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ  
,C;;QAtIhB,OAuiO,W;O;KA9IX,C;4FAUA,yB;MAAA,gE;MAuIA,oC;MAAA,gC;MAvIA,uC;QAOW,kBAaA,eA  
Aa,gBAAb,C;QAwIP,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,0B;UACI,WAAy,WazIwB,SAyIpB,EAAU,c  
AAV,EAAU,sBAAV,WAAmB,iBAAnB,CAAJ,C;;QAzIhB,OA0IO,W;O;KAjJX,C;wGAUA,yB;MAAA,+D;MAA  
A,uC;QAOW,kBAAoB,gB;QA8iEd,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UApiEmC,U;UAAA,cAVQ,S  
AUR,EAoiET,cApiES,EAoiET,sBApiES,WAoiEA,IApiEA,W;YAA6C,6B;;QAVhF,OAwo,W;O;KAlBX,C;4GA  
UA,yB;MAAA,oD;QA2iEiB,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UApiEmC,U;UAAA,yBAoiET,cApiE  
S,EAoiET,sBApiES,WAoiEA,IApiEA,W;YAA6C,6B;;QACHF,OAAO,W;O;KARX,C;8FAWA,6C;MAQiB,UACi







AAV,CAAJ,C;YAAwB,qB;;QAC9C,OAAO,K;O;KANX,C;8EASA,yC;MAUoB,Q;MADhB,kBAAkB,O;MACIB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OA AO,W;K;gFAGX,yC;MAUoB,Q;MADhB,kBAAkB,O;MACIB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;Q AAsB,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;gFAGX,yC;MAUoB,Q;MADhB,kBAAkB, O;MACIB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;M ACpC,OAAO,W;K;gFAGX,yC;MAUoB,Q;MADhB,kBAAkB,O;MACIB,wBAAgB,SAAhB,gB;QAAGB,cAAA,S AAhB,M;QAAsB,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;gFAGX,yC;MAUoB,Q;MADh B,kBAAkB,O;MACIB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,UAAU,WAAV,EAAuB,O AA vB,C;;MACpC,OAAO,W;K;gFAGX,yC;MAUoB,Q;MADhB,kBAAkB,O;MACIB,wBAAgB,SAAhB,gB;QAA gB,cAAA,SAAhB,M;QAAsB,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;gFAGX,yC;MAUo B,Q;MADhB,kBAAkB,O;MACIB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,UAAU,WAAV ,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;gFAGX,yC;MAUoB,Q;MADhB,kBAAkB,O;MACIB,wBAAgB,SAAh B,gB;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;gFAGX, yB;MAAA,oC;MAAA,gC;MAAA,gD;QAUoB,Q;QADhB,kBAAkB,O;QACIB,wBAAgB,SAAhB,gB;UAAgB,cA AhB,UAAgB,SAAhB,O;UAAsB,cAAc,UAAU,WAAV,EAAuB,oBAAvB,C;;QACpC,OAAO,W;O;KAXX,C;4FAc A,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;MACIB,wBAAgB,SAAhB,gB;QAAGB,cAAA,S AAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAA nB,EAAGC,OAAhC,C;;MACpC,OAAO,W ;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;MACIB,wBAAgB,SAAhB,gB;QAAGB, cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAA nB,EAAGC,OAAhC,C;;MACpC, OAAO,W;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;MACIB,wBAAgB,SAAhB,g B;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAA nB,EAAGC,OAAhC,C ;MACpC,OAAO,W;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;MACIB,wBAAgB, SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAA nB,EAAGC,O AAhC,C;;MACpC,OAAO,W;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;MACIB,w BAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAA nB,EAAGC,OAAhC,C;;MACpC,OAAO,W;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kB AAkB,O;MACIB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV, WAAmB,WAA nB,EAAGC,OAAhC,C;;MACpC,OAAO,W;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kB AAkB,O;MACIB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV, WAAmB,WAA nB,EAAGC,OAAhC,C;;MACpC,OAAO,W;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kB AAkB,O;MACIB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV, WAAmB,WAA nB,EAAGC,OAAhC,C;;MACpC,OAAO,W;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kB AAkB,O;MACIB,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV, WAAmB,WAA nB,EAAGC,OAAhC,C;;MACpC,OAAO,W;K;8FAGX,yB;MAAA,oC;MAAA,gC;MAA A,gD;QAYoB,UAA8B,M;QAF9C,YAAY,C;QACZ,kBAAkB,O;QACIB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,U AAgB,SAAhB,O;UAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAA nB,EAAGC,oBAAhC,C;;QACpC, OAAO,W;O;KAbX,C;wFagBA,yB;MAAA,8D;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,kBAAkB,O;Q ACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,YAAJ,EAAL,oBAAJ,OAAV,EAAwB,WAAxB,C;;QAE IB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,8D;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,kBAAkB,O; QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,YAAJ,EAAL,oBAAJ,OAAV,EAAwB,WAAxB,C;; QAEIB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,8D;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,kBA AkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,YAAJ,EAAL,oBAAJ,OAAV,EAAwB,WAAx B,C;;QAEIB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,8D;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,kB AAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,YAAJ,EAAL,oBAAJ,OAAV,EAAwB,WAA xB,C;;QAEIB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,8D;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,k BAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,YAAJ,EAAL,oBAAJ,OAAV,EAAwB,WA AxB,C;;QAEIB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,8D;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,k BAAkB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,YAAJ,EAAL,oBAAJ,OAAV,EAAwB,WA



;MAFb,IAprLO,qBAAQ,CAorLf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmB11aG,MAAO,KnB01aE,GmB11aF,EnB01aO,CmB11aP,C;MnB41ad,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IAtsLO,qBAAQ,CAssLf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;MAEvB,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IAhtLO,qBAAQ,CAGtLf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;MAEvB,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IA1tLO,qBAAQ,CA0tLf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;MAEvB,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IApuLO,qBAAQ,CAouLf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;MAEvB,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IA9uLO,qBAAQ,CA8uLf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,oBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;MAEvB,OAAO,G;K;IAGX,0B;MAaiB,Q;MAFb,IA1vLO,qBAAQ,CA0vLf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmBx8aG,MAAO,KnBw8aE,GmBx8aF,EnBw8aO,CmBx8aP,C;MnB08ad,OAAO,G;K;IAGX,0B;MAaiB,Q;MAFb,IAtwLO,qBAAQ,CAswLf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;MAEvB,OAAO,G;K;gFAGX,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IA51LO,qBAAQ,CA41Lf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAXBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IA2LO,qBAAQ,CA+2Lf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAXBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IA4LO,qBAAQ,CAk4Lf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAXBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IAr5LO,qBAAQ,CAq5Lf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAXBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IAx6LO,qBAAQ,CAw6Lf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAXBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IA37LO,qBAAQ,CA27Lf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAXBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IA98LO,qBAAQ,CA88Lf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C

CV,WAAW,C;;;QAGnB,OAAO,O;O;KAXBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IAj+LO,q  
BAAQ,CAi+Lf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IA  
AI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI  
,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;  
YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAXBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,oC;MAAA,sC;Q  
AWI,IAp/LO,qBAAQ,CAo/Lf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK  
,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,C;QACf,aAAU,CAAV,OAAa,  
SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;Y  
ACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAXBX,C;4FA2BA,yB;MAAA,8D;MAAA,sC;QAOI,IA3  
kMO,qBAAQ,CA2kMf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACr  
B,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;  
UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAA  
U,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA11MO,qBA  
AQ,CA0Imf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cA  
Aa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAA  
Q,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV  
,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAzmMO,qBAAQ,CAym  
Mf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAj  
B,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK  
,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,  
C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAxnMO,qBAAQ,CAwnMf,C;UA  
Ae,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAA  
oB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;  
UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGn  
B,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAvoMO,qBAAQ,CAuoMf,C;UAAe,OAAO,I  
;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O  
;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAA  
Q,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;  
O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAtpMO,qBAAQ,CAspMf,C;UAAe,OAAO,I;QACtB,cA  
Ac,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,e  
AAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,  
CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApB  
X,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IArqMO,qBAAQ,CAqqMf,C;UAAe,OAAO,I;QACtB,cAAc,UAA  
K,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SA  
AS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;  
UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAu  
BA,yB;MAAA,8D;MAAA,sC;QAOI,IAprMO,qBAAQ,CAorMf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C  
;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C  
;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI  
,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAA  
A,8D;MAAA,oC;MAAA,sC;QAOI,IAmsMO,qBAAQ,CamsMf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;  
QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,  
C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI  
I,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MA  
AA,sE;MAAA,8D;MmB17bA,iB;MnBk7bA,sC;QAEiB,Q;QAFb,IAhyMO,qBAAQ,CAGyMf,C;UAAe,MAAM,6B;  
QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACf,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CA  
AL,CAAT,C;UACR,WmB37bG,MAAO,KnB27bO,QmB37bP,EnB27biB,CmB37bjB,C;;QnB67bd,OAAO,Q;O;KA  
nBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBx8bA,iB;MnBw8bA,sC;QAEiB,Q;QAFb,IA9yMO,qBAAQ,CA8y



Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBj9bG,MAAO,KnBi9bO,QmBj9bP,EnBi9biB,CmBj9bjB,C;;QnBm9bd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmB99bA,iB;MnB89bA,sC;QAeiB,Q;QAFb,IA5zMO,qBAAQ,CA4zmf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBv+bG,MAAO,KnBu+bO,QmBv+bP,EnBu+biB,CmBv+bjB,C;;QnBy+bd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBp/bA,iB;MnBo/bA,sC;QAeiB,Q;QAFb,IA10MO,qBAAQ,CA00Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB7/bG,MAAO,KnB6/bO,QmB7/bP,EnB6/biB,CmB7/bjB,C;;QnB+/bd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmB1gcA,iB;MnB0gcA,sC;QAeiB,Q;QAFb,IAx1MO,qBAAQ,CAw1Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBnhcG,MAAO,KnBmhcO,QmBnhcP,EnBmhcib,CmBnhcjB,C;;QnBqhcd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBhicA,iB;MnBgicA,sC;QAeiB,Q;QAFb,IA2MO,qBAAQ,CAs2Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBzicG,MAAO,KnByicO,QmBzicP,EnByicib,CmBzicjB,C;;QnB2icd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBtjcA,iB;MnBsjaA,sC;QAeiB,Q;QAFb,IAp3MO,qBAAQ,CAo3Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB/jcG,MAAO,KnB+jcO,QmB/jcP,EnB+jcib,CmB/jcjB,C;;QnBikcd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmB5kcA,iB;MnB4kcA,sC;QAeiB,Q;QAFb,IA14MO,qBAAQ,Cak4Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBrlcG,MAAO,KnBqlcO,QmBrlcP,EnBqlcib,CmBrlcjB,C;;QnBulcd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MmBlmcA,iB;MnBkmcA,sC;QAeiB,Q;QAFb,IAh5MO,qBAAQ,Cag5Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WmB3mcG,MAAO,KnB2mcO,QmB3mcP,EnB2mciB,CmB3mcjB,C;;QnB6mcd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBnocA,iB;MnBmocA,sC;QAeiB,Q;QAFb,IA+MO,qBAAQ,CAs+Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB5ocG,MAAO,KnB4ocO,QmB5ocP,EnB4ociB,CmB5ocjB,C;;QnB8ocd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBzpcA,iB;MnBypcA,sC;QAeiB,Q;QAFb,IAp/MO,qBAAQ,CAo/Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBlqcG,MAAO,KnBkqcO,QmBlqcP,EnBkqcib,CmBlqcjB,C;;QnBoqcd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmB/qcA,iB;MnB+qcA,sC;QAeiB,Q;QAFb,IAIlgNO,qBAAQ,CAkgNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBxrcG,MAAO,KnBwrcO,QmBxrcP,EnBwrcib,CmBxrcjB,C;;QnB0rcd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmBrscA,iB;MnBqscA,sC;QAeiB,Q;QAFb,IAhhNO,qBAAQ,CaghNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB9scG,MAAO,KnB8scO,QmB9scP,EnB8sciB,CmB9scjB,C;;QnBgtd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmB3tcA,iB;MnB2tcA,sC;QAeiB,Q;QAFb,IA9hNO,qBAAQ,CA8hNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBpucG,MAAO,KnBoucO,QmBpucP,EnBouciB,CmBpucjB,C;;QnBsucd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmBjvcA,iB;MnBivcA,sC;QAeiB,Q;QAFb,IA5iNO,qBAAQ,CA4iNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB1vcG,MAAO,KnB0vcO,QmB1vcP,EnB0vciB,CmB1vcjB,C;;QnB4vcd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmBvwca,iB;MnBuwca,sC;QAeiB,Q;QAFb,IA1jNO,qBAAQ,CA0jNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBhxcG,MAAO,KnBgxcO,QmBhxcP,EnBgxcib,CmBhxcjB,C;;QnBkxcd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmB7xca,iB;MnB6xca,sC;QAeiB,Q;QAFb,IAxkNO,qBAA

Q,CAwkNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBtycG,MAAO,KnBsyscO,QmBtycP,EnBsysciB,CmBtycjB,C;;QnBwycd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MmBnzcA,iB;MnBmzcA,sC;QAeiB,Q;QAFb,IAtlNO,qBAAQ,CAslNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WmB5zcG,MAAO,KnB4zcO,QmB5zcP,EnB4zciB,CmB5zciB,C;;QnB8zcd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA1qNO,qBAAQ,CA0qNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAxrNO,qBAAQ,CAwrNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAtsNO,qBAAQ,CAsNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAptNO,qBAAQ,CAotNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAluNO,qBAAQ,CAkuNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAhvNO,qBAAQ,CAGvNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA9vNO,qBAAQ,CA8vNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA5wNO,qBAAQ,CA4wNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA1xNO,qBAAQ,CA0xNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;4FAsBA,yB;MAAA,8D;MmBpgdA,iB;MnBogdA,sC;QAaiB,Q;QAFb,IAh3NO,qBAAQ,CAG3Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB3gdG,MAAO,KnB2gdO,QmB3gdP,EnB2gdiB,CmB3gdjB,C;;QnB6gdd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBxhdA,iB;MnBwhdA,sC;QAaiB,Q;QAFb,IA53NO,qBAAQ,CA43Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB/hdG,MAAO,KnB+hdO,QmB/hdP,EnB+hdiB,CmB/hdjB,C;;QnBiidd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB5idA,iB;MnB4idA,sC;QAaiB,Q;QAFb,IAx4NO,qBAAQ,CAw4Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBnjdG,MAAO,KnBmjdo,QmBnjdP,EnBmjdiB,CmBnjdjB,C;;QnBqjdd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBhkdA,iB;MnBgkdA,sC;QAaiB,Q;QAFb,IAp5NO,qBAAQ,CAo5Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBvkdG,MAAO,KnBukdo,QmBvkdP,EnBukdiB,CmBvkdjB,C;;QnBykdd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBpldA,iB;MnBoldA,sC;QAaiB,Q;QAFb,IAh6NO,qBAAQ,CAG6Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB3ldG,MAAO,KnB2ldO,QmB3ldP,EnB2ldiB,CmB3ldjB,C;;QnB6ldd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBxmdA,iB;MnBwmdA,sC;QAaiB,Q;QAFb,IA56NO,qB

AAQ,CA46Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB ;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB/mdG,MAAO,KnB+mdO,QmB/mdP,EnB+mdiB,CmB/ mdjB,C;;QnBindd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB5ndA,iB;MnB4ndA,sC;QAaiB,Q;QAFb,I Ax7NO,qBAAQ,CAw7Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAA U,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBnodG,MAAO,KnBmodO,QmBnodP,EnB modiB,CmBnodjB,C;;QnBqodd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBhpdA,iB;MnBgpda,sC;QAa iB,Q;QAFb,IAp8NO,qBAAQ,CAo8Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B; QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBvdpG,MAAO,KnBupdO,QmB vpdP,EnBupdiB,CmBvpdjB,C;;QnBypdd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,oC;MAAA,8D;MmBppqA,i B;MnBoqdA,sC;QAaiB,Q;QAFb,IAh9NO,qBAAQ,Cag9Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAA L,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WmB3qdG, MAAO,KnB2qdO,QmB3qdP,EnB2qdiB,CmB3qdiB,C;;QnB6qdd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D; MmBnsdA,iB;MnBmsdA,sC;QAaiB,Q;QAFb,IApiOO,qBAAQ,CAoiOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,U AAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,W mB1sdG,MAAO,KnB0sdO,QmB1sdP,EnB0sdiB,CmB1sdjB,C;;QnB4sdd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MA AA,8D;MmBvtdA,iB;MnButdA,sC;QAaiB,Q;QAFb,IAhjOO,qBAAQ,CagjOf,C;UAAe,OAAO,I;QACtB,eAAe,SA AS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UA CR,WmB9tdG,MAAO,KnB8tdO,QmB9tdP,EnB8tdiB,CmB9tdjB,C;;QnBgudd,OAAO,Q;O;KAjBX,C;+FAoBA,yB ;MAAA,8D;MmB3udA,iB;MnB2udA,sC;QAaiB,Q;QAFb,IA5jOO,qBAAQ,CA4jOf,C;UAAe,OAAO,I;QACtB,eA Ae,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT, C;UACR,WmBlvdG,MAAO,KnBkvdO,QmBlvdP,EnBkvdjB,CmBlvdjB,C;;QnBovdd,OAAO,Q;O;KAjBX,C;+FAo BA,yB;MAAA,8D;MmB+vdA,iB;MnB+vdA,sC;QAaiB,Q;QAFb,IAxkOO,qBAAQ,CAwkOf,C;UAAe,OAAO,I;QA CtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL, CAAT,C;UACR,WmBtwdG,MAAO,KnBswdO,QmBtwdP,EnBswdiB,CmBtwdjB,C;;QnBwwdd,OAAO,Q;O;KAjB X,C;+FAoBA,yB;MAAA,8D;MmBnxdA,iB;MnBmxdA,sC;QAaiB,Q;QAFb,IAplOO,qBAAQ,CAolOf,C;UAAe,O AAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,U AAK,CAAL,CAAT,C;UACR,WmB1xdG,MAAO,KnB0xdO,QmB1xdP,EnB0xdiB,CmB1xdjB,C;;QnB4xdd,OAAO, Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmBvydA,iB;MnBuydA,sC;QAaiB,Q;QAFb,IAhmOO,qBAAQ,CagmOf ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB9ydG,MAAO,KnB8ydO,QmB9ydP,EnB8ydiB,CmB9ydjB,C;;QnBgz dd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmB3zdA,iB;MnB2zdA,sC;QAaiB,Q;QAFb,IA5mOO,qBAA Q,CA4mOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;U ACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB10dG,MAAO,KnBk0dO,QmB10dP,EnBk0diB,CmB10djB, C;;QnBo0dd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmB/0dA,iB;MnB+0dA,sC;QAaiB,Q;QAFb,IAxnO O,qBAAQ,CAwnOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CA AV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBt1dG,MAAO,KnBs1dO,QmBt1dP,EnBs1diB,C mBt1djB,C;;QnBw1dd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,oC;MAAA,8D;MmBn2dA,iB;MnBm2dA,sC; QAaiB,Q;QAFb,IApoOO,qBAAQ,CAooOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QAC F,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WmB12dG,MAAO,KnB02dO ,QmB12dP,EnB02diB,CmB12djB,C;;QnB42dd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWi B,Q;QAFb,IAAttOO,qBAAQ,CAstOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;Q AAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI, WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IALuOO,qBAA Q,CAkuOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;U ACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,WAAW,C;;;QAGnB,OA AO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA9uOO,qBAAQ,CA8uOf,C;UAAe,OA AO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAA K,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FA

oBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA1vOO,qBAAQ,CA0vOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAtwOO,qBAAQ,CAswOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAIxOO,qBAAQ,CakxOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA9xOO,qBAAQ,CA8xOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA1yOO,qBAAQ,CA0yOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,oC;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAtzOO,qBAAQ,CAszOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;wFAoBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IA54OO,qBAAQ,CA44Of,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KANBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IA15OO,qBAAQ,CA05Of,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KANBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAx6OO,qBAAQ,CAw6Of,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KANBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IA7OO,qBAAQ,CAs7Of,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KANBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAp8OO,qBAAQ,CAo8Of,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KANBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAI9OO,qBAAQ,CAk9Of,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KANBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAh+OO,qBAAQ,CAg+Of,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KANBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IA9+OO,qBAAQ,CA8+Of,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KANBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IA5/OO,qBAAQ,CA4/Of,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KANBX,C;oGAsBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAhIPO,qBAAQ,CAGlPf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SA

AS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI, WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA5IPO,qBAA Q,CA4IPf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UA CI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,C AAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb, IAxmPO,qBAAQ,CAwmPf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aA AU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB, CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD; QAWiB,Q;QAFb,IApnPO,qBAAQ,CAonPf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QAC F,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,E AakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA, 8D;MAAA,kD;QAWiB,Q;QAFb,IAhoPO,qBAAQ,CAgoPf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL, CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,S AAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAo BA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA5oPO,qBAAQ,CA4oPf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS ,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR, IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;K AjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAxpPO,qBAAQ,CAwpPf,C;UAAe,OAAO,I;QAC tB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,C AAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGn B,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IApqPO,qBAAQ,CAoqPf,C;UAA e,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS, UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,W AAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,oC;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAhrP O,qBAAQ,CAgrPf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAA V,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,G AakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;IAoBA,8B;MASiB,Q;MAFb,IALwPO,qBAAQ ,CAkwPf,C;QAae,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,U AAK,CAAL,C;QACR,MmB75eG,MAAO,KnB65eE,GmB75eF,EnB65eO,CmB75eP,C;;;MnB+5ed,OAAO,G;K;IA GX,gC;MASiB,Q;MAFb,IALxPO,qBAAQ,CAkxPf,C;QAae,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,+B ;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmBx7eG,MAAO,KnBw7eE,GmBx7eF,EnBw7e O,CmBx7eP,C;;;MnB07ed,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IAhyPO,qBAAQ,CAGyPf,C;QAae,OAAO,I;M ACTB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,s BAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IAtyPO,qBAAQ,CAs yPf,C;QAae,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK, CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IA5 yPO,qBAAQ,CA4yPf,C;QAae,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QA CI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MA OiB,Q;MAFb,IALzPO,qBAAQ,CAkzPf,C;QAae,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aA AU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO, G;K;IAGX,gC;MAOiB,Q;MAFb,IAxzPO,qBAAQ,CAwzPf,C;QAae,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;M ACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,oBAAM,CAAN,KAAJ,C;UAAa,M AAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MASiB,Q;MAFb,IAh0PO,qBAAQ,CAG0Pf,C;QAae,OAAO,I;MACTB,U AAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmB9gfG,MA AO,KnB8gfE,GmB9gfF,EnB8gfO,CmB9gfP,C;;;MnBghfd,OAAO,G;K;IAGX,gC;MASiB,Q;MAFb,IAx0PO,qBAA Q,CAw0Pf,C;QAae,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ, UAAK,CAAL,C;QACR,MmBnhfG,MAAO,KnBmhfE,GmBnhfF,EnBmhfO,CmBnhfP,C;;;MnBqhfd,OAAO,G;K;I AGX,gC;MAOiB,Q;MAFb,IAt0PO,qBAAQ,CAs0Pf,C;QAae,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,i

C;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAE vB,OAAO,G;K;IAGX,wC;MAWiB,Q;MAFb,IAx5PO,qBAAQ,CAw5Pf,C;QAAe,MAAM,6B;MACrB,UAAU,UA AK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GA AR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAF b,IAI6PO,qBAAQ,CAk6Pf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAA V,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C; UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IA56PO,qBAAQ,CA46Pf,C;QAAe,MAAM ,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,I AAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IA GX,0C;MAWiB,Q;MAFb,IAf7PO,qBAAQ,CAs7Pf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG ,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CA AX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IAh8PO,qBAAQ,C Ag8Pf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,U AAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;; MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IA18PO,qBAAQ,CA08Pf,C;QAAe,MAAM,6B;MACrB,UAAU, UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ, GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;M AFb,IAp9PO,qBAAQ,CAo9Pf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,C AAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC ,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IA99PO,qBAAQ,CA89Pf,C;QAAe,MA AM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QAC R,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;I AGX,0C;MAWiB,Q;MAFb,IAx+PO,qBAAQ,CAw+Pf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MA CG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,gBAAR,EAAa,cAAb, CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,8C;MAOiB,Q;MAFb,IAtjQO,qBAAQ, CAsjQf,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,U AAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;; MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IA5jQO,qBAAQ,CA4jQf,C;QAAe,OAAO,I;MACTB,UAAU,UA AK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GA AR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb ,IAlkQO,qBAAQ,CAkkQf,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB ;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAA oC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAxkQO,qBAAQ,CAwkQf,C;QAAe,OAAO,I;M ACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,U AAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD; MAOiB,Q;MAFb,IA9kQO,qBAAQ,CA8kQf,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAA b,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA 6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAplQO,qBAAQ,CAolQf,C;Q AAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL, C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAA O,G;K;IAGX,gD;MAOiB,Q;MAFb,IA1lQO,qBAAQ,CA0lQf,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C; MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CA Ab,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAhmQO,qB AAQ,CAgmQf,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QA AQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAA M,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAtmQO,qBAAQ,CAsmQf,C;QAAe,OAAO,I;MACTB,UA AU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SA AQ,gBAAR,EAAa,cAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,wB;MAaiB,

Q;MAFb,IA1rQO,qBAAQ,CA0rQf,C;QA Ae,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmBjofG,MAAO,KnBiofE,GmBjofF,EnBiofO,CmBjofP,C;;MnBmofd,OAAO,G;K;IAGX,0B;MAaiB,Q;MAFb,IA9sQO,qBAAQ,CA8sQf,C;QA Ae,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmBhqfG,MAAO,KnBgqfE,GmBhqfF,EnBgqfO,CmBhqfP,C;;MnBkqfd,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IAhuQO,qBAAQ,CAguQf,C;QA Ae,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IA1uQO,qBAAQ,CA0uQf,C;QA Ae,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IApvQO,qBAAQ,CAovQf,C;QA Ae,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IA9vQO,qBAAQ,CA8vQf,C;QA Ae,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IAxwQO,qBAAQ,CAwwQf,C;QA Ae,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,oBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAaiB,Q;MAFb,IApxQO,qBAAQ,CAoxQf,C;QA Ae,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmB9wfG,MAAO,KnB8wfE,GmB9wfF,EnB8wfO,CmB9wfP,C;;MnBgxf d,OAAO,G;K;IAGX,0B;MAaiB,Q;MAFb,IAhyQO,qBAAQ,CAgyQf,C;QA Ae,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmBvxfG,MAAO,KnBuxfE,GmBvxfF,EnBuxfO,CmBvxfP,C;;MnByxfd,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IAlyQO,qBAAQ,CAkyQf,C;QA Ae,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;gFAGX,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IA13QO,qBAAQ,CA3Qf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IA24QO,qBAAQ,CAy4Qf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IA55QO,qBAAQ,CA45Qf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IA60QO,qBAAQ,CA+6Qf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IA89QO,qBAAQ,CA89Qf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IAx+QO,qBAAQ,CAw+Qf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C

;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C  
;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI  
,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;KAXBX,C;kFA2BA,yB;MAA  
A,sE;MAAA,8D;MAAA,sC;QAWI,IA3/QO,qBAAQ,CA2/Qf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C  
;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C  
;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI  
,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;KAXBX,C;kFA2BA,yB;MAA  
A,sE;MAAA,8D;MAAA,oC;MAAA,sC;QAWI,IA9gRO,qBAAQ,CA8gRf,C;UAAe,MAAM,6B;QACrB,cAAc,UA  
AK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,S  
AAS,oBAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,  
C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;KAXBX,C;4F  
A2BA,yB;MAAA,8D;MAAA,sC;QAOI,IArmRO,qBAAQ,CAqmRf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAA  
L,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAA  
T,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,I  
AAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;KApBX,C;8FAuBA,yB;  
MAAA,8D;MAAA,sC;QAOI,IApnRO,qBAAQ,CAonRf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,  
gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,  
aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAA  
W,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;KApBX,C;8FAuBA,yB;MAAA,8D;  
MAAA,sC;QAOI,IAnoRO,qBAAQ,CAmoRf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,c  
AAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CA  
AV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,  
KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;  
QAOI,IAIpRO,qBAAQ,CAkpRf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,  
C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,S  
AAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YA  
CI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAjq  
RO,qBAAQ,CAiqRf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,I  
AAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UA  
CI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,  
C;YACV,WAAW,C;;;QAGnB,OAAO,O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAhrRO,qBAAQ  
,CAgrRf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,C  
AAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,U  
AAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WA  
AW,C;;;QAGnB,OAAO,O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA/rRO,qBAAQ,CA+rRf,C;U  
AAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UA  
AoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,  
C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QA  
GnB,OAAO,O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA9sRO,qBAAQ,CA8sRf,C;UAAe,OAAO  
,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,  
O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QA  
AQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,  
O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,oC;MAAA,sC;QAOI,IA7tRO,qBAAQ,CA6tRf,C;UAAe,OAAO,I  
;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O  
;QAC3B,eAAe,SAAS,oBAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QA  
AQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O  
;KApBX,C;gFAuBA,yB;MAAA,sE;MAAA,8D;MmBxvgBA,iB;MnBwvgBA,sC;QAEiB,Q;QAFb,IA1zRO,qBA  
AQ,CA0zRf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,i



B;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBjwgBG,MAAO,KnBiwgBO,QmBjwgBP,EnBiwgBiB,CmBjwgBjB,C;;QnBmwgBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmB9wgBA,iB;MnB8wgBA,sC;QAeiB,Q;QAFb,IAx0RO,qBAAQ,CAw0Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBvvgBG,MAAO,KnBuxgBO,QmBvvgBP,EnBuxgBiB,CmBvvgBjB,C;;QnByxgBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBpygBA,iB;MnBoygBA,sC;QAeiB,Q;QAFb,IA1RO,qBAAQ,CAs1Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB7ygBG,MAAO,KnB6ygBO,QmB7ygBP,EnB6ygBiB,CmB7ygBjB,C;;QnB+ygBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmB1zgBA,iB;MnB0zgBA,sC;QAeiB,Q;QAFb,IAp2RO,qBAAQ,CAo2Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBn0gBG,MAAO,KnBm0gBO,QmBn0gBP,EnBm0gBiB,CmBn0gBjB,C;;QnBq0gBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBh1gBA,iB;MnB1gBA,sC;QAeiB,Q;QAFb,IA13RO,qBAAQ,CAk3Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBz1gBG,MAAO,KnBy1gBO,QmBz1gBP,EnBy1gBiB,CmBz1gBjB,C;;QnB21gBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBt2gBA,iB;MnBs2gBA,sC;QAeiB,Q;QAFb,IAh4RO,qBAAQ,CAg4Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB/2gBG,MAAO,KnB+2gBO,QmB/2gBP,EnB+2gBiB,CmB/2gBjB,C;;QnBi3gBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmB53gBA,iB;MnB43gBA,sC;QAeiB,Q;QAFb,IA94RO,qBAAQ,CA84Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBr4gBG,MAAO,KnBq4gBO,QmBr4gBP,EnBq4gBiB,CmBr4gBjB,C;;QnBu4gBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBl5gBA,iB;MnBk5gBA,sC;QAeiB,Q;QAFb,IA55RO,qBAAQ,CA45Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB35gBG,MAAO,KnB25gBO,QmB35gBP,EnB25gBiB,CmB35gBjB,C;;QnB65gBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MmBx6gBA,iB;MnBw6gBA,sC;QAeiB,Q;QAFb,IA16RO,qBAAQ,CA06Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WmBj7gBG,MAAO,KnBi7gBO,QmBj7gBP,EnBi7gBiB,CmBj7gBjB,C;;QnBm7gBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBz8gBA,iB;MnBy8gBA,sC;QAeiB,Q;QAFb,IAhgSO,qBAAQ,CAggSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBl9gBG,MAAO,KnBk9gBO,QmBl9gBP,EnBk9gBiB,CmBl9gBjB,C;;QnBo9gBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmB/9gBA,iB;MnB+9gBA,sC;QAeiB,Q;QAFb,IA9gSO,qBAAQ,CA8gSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBx+gBG,MAAO,KnBw+gBO,QmBx+gBP,EnBw+gBiB,CmBx+gBjB,C;;QnB0+gBd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmBr/gBA,iB;MnBq/gBA,sC;QAeiB,Q;QAFb,IA5hSO,qBAAQ,CA4hSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB9/gBG,MAAO,KnB8/gBO,QmB9/gBP,EnB8/gBiB,CmB9/gBjB,C;;QnBgghBd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmB3ghBA,iB;MnB2ghBA,sC;QAeiB,Q;QAFb,IA1iSO,qBAAQ,CA0iSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBphhBG,MAAO,KnBohhBO,QmBphhBP,EnBohhBiB,CmBphhBjB,C;;QnBshhBd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmBjihBA,iB;MnBiuhBA,sC;QAeiB,Q;QAFb,IAxjSO,qBAAQ,CAwjSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB1ihBG,MAAO,KnB0ihBO,QmB1ihBP,EnB0ihBiB,CmB1ihBjB,C;;QnB4ihBd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmBvjhBA,iB;MnBujhBA,sC;QAeiB,Q;QAFb,IAtkSO,qBAAQ,CAskSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBhkhBG,MAAO,KnBgkhBO,QmBhkhBP,EnBgkhBiB,CmBhkhBjB,C;;QnBkhhBd,OAAO,Q;O;KAnBX,C;mF

AsBA,yB;MAAA,sE;MAAA,8D;MmB7khBA,iB;MnB6khBA,sC;QAeiB,Q;QAFb,IAplSO,qBAAQ,CAolSf,C;UA Ae,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SA AS,UAAK,CAAL,CAAT,C;UACR,WmBtlhBG,MAAO,KnBslhBO,QmBtlhBP,EnBslhBiB,CmBtlhBjB,C;;QnBwl hBd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmBnmhBA,iB;MnBmmhBA,sC;QAeiB,Q;QAF b,IAImSO,qBAAQ,CAkmSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb, aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB5mhBG,MAAO,KnB4mhBO,QmB5m hBP,EnB4mhBiB,CmB5mhBjB,C;;QnB8mhBd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,oC;MAA A,8D;MmBznhBA,iB;MnBznhBA,sC;QAeiB,Q;QAFb,IAhnSO,qBAAQ,CagnSf,C;UAAe,MAAM,6B;QACrB,eA Ae,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT ,C;UACR,WmBlbhBG,MAAO,KnBkohBO,QmBlbhBP,EnBkohBiB,CmBlbhBjB,C;;QnBoohBd,OAAO,Q;O;KAn BX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IApsSO,qBAAQ,CAosSf,C;UAAe,MAAM, 6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK, CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAs BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAItSO,qBAAQ,CAktSf,C;UAAe,MAAM,6B;QACrB,e AAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAA T,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAA A,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAhuSO,qBAAQ,CaguSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS, UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,I AAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAA A,8D;MAAA,sC;QAaiB,Q;QAFb,IA9uSO,qBAAQ,CA8uSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CA AL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAA W,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAA A,sC;QAaiB,Q;QAFb,IA5vSO,qBAAQ,CA4vSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT, C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX, KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAai B,Q;QAFb,IA1wSO,qBAAQ,CA0wSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+ B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;Y ACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb, IAxxSO,qBAAQ,CAwxSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,a AAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAA W,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAtySO,q BAAQ,CAsySf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAA V,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QA GnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IApzSO, qBAAQ,CAozSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAA V,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;Q AGnB,OAAO,Q;O;KAnBX,C;4FAsBA,yB;MAAA,8D;MmB10hBA,iB;MnB00hBA,sC;QAaiB,Q;QAFb,IA14SO,q BAAQ,CA04Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,i B;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBj1hBG,MAAO,KnBi1hBO,QmBj1hBP,EnBi1hBiB,C mBj1hBjB,C;;QnBm1hBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB91hBA,iB;MnB81hBA,sC;QAaiB ,Q;QAFb,IAt5SO,qBAAQ,CAs5Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QA Ab,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBr2hBG,MAAO,KnBq2hBO,QmBr 2hBP,EnBq2hBiB,CmBr2hBjB,C;;QnBu2hBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB13hBA,iB;Mn Bk3hBA,sC;QAaiB,Q;QAFb,IAI6SO,qBAAQ,CAk6Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAA T,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBz3hBG,MAA O,KnBy3hBO,QmBz3hBP,EnBy3hBiB,CmBz3hBjB,C;;QnB23hBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8 D;MmBt4hBA,iB;MnBs4hBA,sC;QAaiB,Q;QAFb,IA96SO,qBAAQ,CA86Sf,C;UAAe,OAAO,I;QACtB,eAAe,SA AS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UA

CR,WmB74hBG,MAAO,KnB64hBO,QmB74hBP,EnB64hBiB,CmB74hBjB,C;;QnB+4hBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB15hBA,iB;MnB05hBA,sC;QAaiB,Q;QAFb,IA17SO,qBAAQ,CA07Sf,C;UAAe,OA AO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAA K,CAAL,CAAT,C;UACR,WmBj6hBG,MAAO,KnBi6hBO,QmBj6hBP,EnBi6hBiB,CmBj6hBjB,C;;QnBm6hBd,O AAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB96hBA,iB;MnB86hBA,sC;QAaiB,Q;QAFb,IAt8SO,qBAAQ,C As8Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI, QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBr7hBG,MAAO,KnBq7hBO,QmBr7hBP,EnBq7hBiB,CmBr7h BjB,C;;QnBu7hBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB18hBA,iB;MnBk8hBA,sC;QAaiB,Q;QAF b,IA19SO,qBAAQ,CAk9Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAA U,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBz8hBG,MAAO,KnBy8hBO,QmBz8hBP,E nBy8hBiB,CmBz8hBjB,C;;QnB28hBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBt9hBA,iB;MnBs9hB A,sC;QAaiB,Q;QAFb,IA99SO,qBAAQ,CA89Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C; QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB79hBG,MAAO,Kn B69hBO,QmB79hBP,EnB69hBiB,CmB79hBjB,C;;QnB+9hBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,oC;M AAA,8D;MmB1+hBA,iB;MnB0+hBA,sC;QAaiB,Q;QAFb,IA1+SO,qBAAQ,CA0+Sf,C;UAAe,OAAO,I;QACtB,e AAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EA AT,C;UACR,WmBj/hBG,MAAO,KnBi/hBO,QmBj/hBP,EnBi/hBiB,CmBj/hBjB,C;;QnBm/hBd,OAAO,Q;O;KAjB X,C;8FAoBA,yB;MAAA,8D;MmBzgiBA,iB;MnBygiBA,sC;QAaiB,Q;QAFb,IA9jTO,qBAAQ,CA8jTf,C;UAAe,O AAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UA AK,CAAL,CAAT,C;UACR,WmBhhiBG,MAAO,KnBghiBO,QmBhhiBP,EnBghiBiB,CmBhhiBjB,C;;QnBkhiBd,O AAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB7hiBA,iB;MnB6hiBA,sC;QAaiB,Q;QAFb,IA1kTO,qBAAQ,C A0kTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI, QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBpiiBG,MAAO,KnBoiiBO,QmBpiiBP,EnBoiiBiB,CmBpiiBjB, C;;QnBsiiBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmBjjiBA,iB;MnBijiBA,sC;QAaiB,Q;QAFb,IAtlT O,qBAAQ,CAslTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAA V,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBxjiBG,MAAO,KnBwjiBO,QmBxjiBP,EnBwjiBiB, CmBxjiBjB,C;;QnB0jiBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmBrkiBA,iB;MnBqkiBA,sC;QAaiB,Q ;QAFb,IAImTO,qBAAQ,CAkmTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QA Ab,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB5kiBG,MAAO,KnB4kiBO,QmB5 kiBP,EnB4kiBiB,CmB5kiBjB,C;;QnB8kiBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmBzliBA,iB;MnBy liBA,sC;QAaiB,Q;QAFb,IA9mTO,qBAAQ,CA8mTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAA T,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBhmiBG,MAAO ,KnBgmiBO,QmBhmiBP,EnBgmiBiB,CmBhmiBjB,C;;QnBkmiBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D ;MmB7miBA,iB;MnB6miBA,sC;QAaiB,Q;QAFb,IA1nTO,qBAAQ,CA0nTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR ,WmBpniBG,MAAO,KnBonniBO,QmBpniBP,EnBonniBiB,CmBpniBjB,C;;QnBsniBd,OAAO,Q;O;KAjBX,C;+FAo BA,yB;MAAA,8D;MmBjoiBA,iB;MnBioiBA,sC;QAaiB,Q;QAFb,IAtoTO,qBAAQ,CAsoTf,C;UAAe,OAAO,I;QA CtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL, CAAT,C;UACR,WmBxoiBG,MAAO,KnBwoiBO,QmBxoiBP,EnBwoiBiB,CmBxoiBjB,C;;QnB0oiBd,OAAO,Q;O ;KAjBX,C;+FAoBA,yB;MAAA,8D;MmBrpiBA,iB;MnBqpiBA,sC;QAaiB,Q;QAFb,IAlpTO,qBAAQ,CAkpTf,C;U AAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SA AS,UAAK,CAAL,CAAT,C;UACR,WmB5piBG,MAAO,KnB4piBO,QmB5piBP,EnB4piBiB,CmB5piBjB,C;;QnB8 piBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,oC;MAAA,8D;MmBzqiBA,iB;MnByqiBA,sC;QAaiB,Q;QAFb,I A9pTO,qBAAQ,CA8pTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAA U,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WmBhriBG,MAAO,KnBgriBO,QmBhriBP,En BgriBiB,CmBhriBjB,C;;QnBkriBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,I AhvTO,qBAAQ,CAgvTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU ,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,WAAW,C;;



AIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA ,oC;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAthUO,qBAAQ,CashUf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,s BAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,I AAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;K AnBX,C;oGAsBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA1mUO,qBAAQ,CA0mUf,C;UAAe,OAAO,I;QA CtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL, CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAG nB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAtnUO,qBAAQ,CAsnUf,C;UA Ae,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS ,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,W AAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAloUO,qBAAQ, CAkoUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UAC I,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CA AtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,I A9oUO,qBAAQ,CA8oUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAA U,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,C AAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD; QAWiB,Q;QAFb,IA1pUO,qBAAQ,CA0pUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QAC F,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,E AakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA, 8D;MAAA,kD;QAWiB,Q;QAFb,IAtqUO,qBAAQ,CAsqUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL, CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,S AAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAo BA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAlrUO,qBAAQ,CAkrUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS, UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,I AAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;K AjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA9rUO,qBAAQ,CA8rUf,C;UAAe,OAAO,I;QACt B,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,C AAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGn B,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,oC;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA1sUO,qBAAQ,CA0 sUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,Q AAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAt C,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;IAoBA,8B;MASiB,Q;MAFb,IA5xUO,qBAAQ,CA4xUf,C; QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAA L,C;QACR,MmBnuijBG,MAAO,KnBmujBE,GmBnuijBF,EnBmujBO,CmBnuijBP,C;;MnBquijBd,OAAO,G;K;IAGX ,gC;MASiB,Q;MAFb,IA5yUO,qBAAQ,CA4yUf,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,+B; MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmB9vjBG,MAAO,KnB8vjBE,GmB9vjBF,EnB8 vjBO,CmB9vjBP,C;;MnBgwjBd,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IA1zUO,qBAAQ,CA0zUf,C;QAAe,OA AO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR ,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IAh0UO,qBA AQ,CAg0Uf,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAA Q,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q; MAFb,IAt0UO,qBAAQ,CAs0Uf,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CA AV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IA GX,gC;MAOiB,Q;MAFb,IA50UO,qBAAQ,CA40Uf,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,i C;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAE vB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IA11UO,qBAAQ,CAk1Uf,C;QAAe,OAAO,I;MACTB,UAAU,UAAK, CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,oBAAM,CAAN,KAAJ,

C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MASiB,Q;MAFb,IA11UO,qBAAQ,CA01Uf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmBp1jBG,MAAO,KnBo1jBE,GmBp1jBF,EnBo1jBO,CmBp1jBP,C;;MnBs1jBd,OAAO,G;K;IAGX,gC;MASiB,Q;MAFb,IAI2UO,qBAAQ,CAk2Uf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmBz1jBG,MAAO,KnBy1jBE,GmBz1jBF,EnBy1jBO,CmBz1jBP,C;;MnB21jBd,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IAh2UO,qBAAQ,CAg2Uf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,wC;MAWiB,Q;MAFb,IAI7UO,qBAAQ,CAk7Uf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IA57UO,qBAAQ,CA47Uf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IAI8UO,qBAAQ,CA8Uf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IAI9UO,qBAAQ,CA9Uf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IAp+UO,qBAAQ,CAo+Uf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IA9+UO,qBAAQ,CA8+Uf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IAx/UO,qBAAQ,CAw/Uf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IAIgVO,qBAAQ,CAkgVf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,8C;MAOiB,Q;MAFb,IAh1VO,qBAAQ,CAg1Vf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAI1VO,qBAAQ,CAI1Vf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAI1VO,qBAAQ,CAI1Vf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAxmVO,qBAAQ,CAxmVf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IA9mVO,qBAAQ,CA8mVf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IApnVO,qBAAQ,CAonVf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UA

AW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;M  
AOiB,Q;MAFb,IA1nVO,qBAAQ,CA0nVf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,  
aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6  
B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAhoVO,qBAAQ,CAGoVf,C;Q  
AAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,  
C;QACR,IAAI,UAAW,SAAQ,gBAAR,EAAa,cAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAA  
O,G;K;IAGX,yB;MAMI,OA/sVO,qBAAQ,C;K;IAktVnB,2B;MAMI,OAhtVO,qBAAQ,C;K;IAmtVnB,2B;MAMI,  
OAjtVO,qBAAQ,C;K;IAotVnB,2B;MAMI,OAltVO,qBAAQ,C;K;IAqtVnB,2B;MAMI,OAntVO,qBAAQ,C;K;IAst  
VnB,2B;MAMI,OaptVO,qBAAQ,C;K;IAutVnB,2B;MAMI,OArtVO,qBAAQ,C;K;IAwtVnB,2B;MAMI,OAttVO,  
qBAAQ,C;K;IAytVnB,2B;MAMI,OAvtVO,qBAAQ,C;K;gFA0tVnB,gC;MAMoB,Q;MAAhB,wBAAGB,SAAhB,g  
B;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;gFA  
GX,gC;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ  
,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,  
SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MAMoB,Q;  
MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,  
K;;MACrD,OAAO,I;K;iFAGX,gC;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,  
IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MAMoB,Q;MAAhB,wBAAGB,  
SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,  
I;K;iFAGX,gC;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAA  
V,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MAMoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAG  
B,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,yB;M  
AAA,oC;MAAA,gC;MAAA,uC;QAMoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;U  
AAsB,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,K;;QACrD,OAAO,I;O;KAPX,C;kFAUA,6B;MAMmC,Q;  
MAAhB,iD;QAAGB,cAAhB,e;QAAsB,OAAO,OAAP,C;;MAArC,gB;K;oFAGJ,6B;MAMmC,Q;MAAhB,iD;QAAGB,cAAhB,e;QA  
AsB,OAAO,OAAP,C;;MAArC,gB;K;oFAGJ,6B;MAMmC,Q;MAAhB,iD;QAAGB,cAAhB,e;QAAsB,OAAO,OAAP,  
C;;MAArC,gB;K;oFAGJ,6B;MAMmC,Q;MAAhB,iD;QAAGB,cAAhB,e;QAAsB,OAAO,OAAP,C;;MAArC,gB;  
K;oFAGJ,6B;MAMmC,Q;MAAhB,iD;QAAGB,cAAhB,e;QAAsB,OAAO,OAAP,C;;MAArC,gB;K;oFAGJ,6B;MA  
MmC,Q;MAAhB,iD;QAAGB,cAAhB,e;QAAsB,OAAO,OAAP,C;;MAArC,gB;K;oFAGJ,6B;MAMmC,Q;MAAhB,  
iD;QAAGB,cAAhB,e;QAAsB,OAAO,OAAP,C;;MAArC,gB;K;oFAGJ,yB;MAAA,oC;MAAA,gC;MAAA,oC;QA  
MmC,Q;QAAhB,iD;UAAgB,cAAhB,0B;UAAAsB,OAAO,oBAAP,C;;QAARc,gB;O;KANJ,C;gGASA,6B;MAn4Ki  
B,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAMb,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MA  
44KnB,gB;K;kGAGJ,6B;MAr4KiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAMb,QAAO,cAAP,EAA  
O,sBAAP,WAAgB,IAAhB,C;;MA84KnB,gB;K;kGAGJ,6B;MAv4KiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WA  
Ab,e;QAAMb,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAG5KnB,gB;K;kGAGJ,6B;MAz4KiB,gB;MAD  
b,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAMb,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAk5KnB,gB;  
K;kGAGJ,6B;MA34KiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAMb,QAAO,cAAP,EAAO,sBAAP,  
WAAgB,IAAhB,C;;MAo5KnB,gB;K;kGAGJ,6B;MA74KiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAM  
b,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAS5KnB,gB;K;kGAGJ,6B;MA/4KiB,gB;MADb,YAAY,C  
;MACZ,iD;QAAa,WAAb,e;QAAMb,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAw5KnB,gB;K;kGAGJ,  
6B;MAj5KiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAMb,QAAO,cAAP,EAAO,sBAAP,WAAgB,IA  
AhB,C;;MA05KnB,gB;K;kGAGJ,yB;MAAA,6B;MAAA,sC;MA15KA,oC;MAAA,gC;MA05KA,2BASiB,yB;QAn  
6KjB,oC;QAAA,gC;eAm6KiB,0B;UAAA,4B;YAAE,aAAe,c;YA55KjB,gB;YADb,YAAY,C;YACZ,iD;cAAA,WA  
Ab,0B;cAAMb,QAAO,cAAP,EAAO,sBAAP,WAAgB,iBAAhB,C;;YA45KmB,W;W;S;OAAzB,C;MATjB,oC;QA  
n5KiB,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,0B;UAAmB,QAAO,cAAP,EAAO,sBAAP,WAAgB,iBAAhB,  
C;;QA45KnB,gB;O;KATJ,C;kFAYA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA9jWO,qBAAQ,  
CA8jWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAqB,UAAK,CAAL,C;QACJ,+B;QAAjB,iBAAc,CA  
Ad,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KANBX,C;oFAsBA,y

B;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA5kWO,qBAAQ,CA4kWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA11WO,qBAAQ,CA01Wf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAxmWO,qBAAQ,CAwmWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAtnWO,qBAAQ,CAsnWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IApwoWO,qBAAQ,CAooWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAlpWO,qBAAQ,CAkpWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAhqWO,qBAAQ,CAgqWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA9qWO,qBAAQ,CA8qWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,wBAAV,EAAuB,sBAakB,KAAL,EAA9B,E;;QAEIB,OAAO,W;O;KAnBX,C;gGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IApwoWO,qBAAQ,CAowWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAqB,UAAK,CAAL,C;QACJ,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAav,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAlxWO,qBAAQ,CAkxWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAav,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA9yWO,qBAAQ,CA8yWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAav,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA5zWO,qBAAQ,CA4zWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAav,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA10WO,qBAAQ,CA00Wf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAav,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAx1WO,qBAAQ,CAw1Wf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAav,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA2WO,qBAAQ,CAs2Wf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAav,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAp3WO,qBAAQ,CAo3Wf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,KAav,EAAiB,wBAAjB,EAA8B,sBAakB,KAAL,EAA9B,E;;QAEIB,OAAO,W;O;KAnBX,C;4GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA18WO,qBAAQ,CA08Wf,C



;UACI,OAAO,I;QACX,kBAAqB,UAAK,CAAL,C;QACJ,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAA V,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;M AAA,uC;QAgBqB,Q;QAHjB,IAx9WO,qBAAQ,CAw9Wf,C;UACI,OAAO,I;QACX,kBAaKB,UAAK,CAAL,C;Q ACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B, C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAt+WO,qBAAQ,CA s+Wf,C;UACI,OAAO,I;QACX,kBAaKB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UA AU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MA AAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAp/WO,qBAAQ,CAo/Wf,C;UACI,OAAO,I;QACX,kBAaKB,UAAK,CA AL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,C AA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAIlgXO,qBAA Q,CAkgXf,C;UACI,OAAO,I;QACX,kBAaKB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc ,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB; MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAhhXO,qBAAQ,CAghXf,C;UACI,OAAO,I;QACX,kBAaKB,UAAK, CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAA L,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA9hXO,q BAAQ,CA8hXf,C;UACI,OAAO,I;QACX,kBAaKB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI, cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsB A,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA5iXO,qBAAQ,CA4iXf,C;UACI,OAAO,I;QACX,kBAaKB,U AAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK, KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,oC;MAAA,gC;MAAA,uC;Q AgBqB,Q;QAHjB,IA1jXO,qBAAQ,CA0jXf,C;UACI,OAAO,I;QACX,kBAaKB,UAAK,CAAL,C;QACD,+B;QAA jB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,KAAV,EAAiB,wBAAjB,EAA8B,sBAAK,KAAL,EAA9B,E;;QAEIB,O AAO,W;O;KAnBX,C;8FAsBA,yB;MAAA,8D;MAAA,uC;QAIbqB,Q;QAHjB,IAjpXO,qBAAQ,CAipXf,C;UACI, OAAO,I;QACX,kBAAqB,UAAK,CAAL,C;QACJ,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAA uB,UAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KApBX,C;gGAuBA,yB;MAAA,8D;MAAA,uC;QAIbqB,Q;Q AHjB,IAhqXO,qBAAQ,CAgqXf,C;UACI,OAAO,I;QACX,kBAaKB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc, CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KApBX,C;gGAuB A,yB;MAAA,8D;MAAA,uC;QAIbqB,Q;QAHjB,IA/qXO,qBAAQ,CA+qXf,C;UACI,OAAO,I;QACX,kBAaKB,U AAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB, C;;QAEIB,OAAO,W;O;KApBX,C;gGAuBA,yB;MAAA,8D;MAAA,uC;QAIbqB,Q;QAHjB,IA9rXO,qBAAQ,CA8 rXf,C;UACI,OAAO,I;QACX,kBAaKB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU ,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KApBX,C;gGAuBA,yB;MAAA,8D;MAAA,uC; QAIbqB,Q;QAHjB,IA7sXO,qBAAQ,CA6sXf,C;UACI,OAAO,I;QACX,kBAaKB,UAAK,CAAL,C;QACD,+B;QA AjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KApB X,C;gGAuBA,yB;MAAA,8D;MAAA,uC;QAIbqB,Q;QAHjB,IA5tXO,qBAAQ,CA4tXf,C;UACI,OAAO,I;QACX,k BAaKB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAA L,CAAvB,C;;QAEIB,OAAO,W;O;KApBX,C;gGAuBA,yB;MAAA,8D;MAAA,uC;QAIbqB,Q;QAHjB,IA3uXO,q BAAQ,CA2uXf,C;UACI,OAAO,I;QACX,kBAaKB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI, cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KApBX,C;gGAuBA,yB;MAAA,8D ;MAAA,uC;QAIbqB,Q;QAHjB,IA1vXO,qBAAQ,CA0vXf,C;UACI,OAAO,I;QACX,kBAaKB,UAAK,CAAL,C;Q ACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAEIB,OAA O,W;O;KApBX,C;gGAuBA,yB;MAAA,8D;MAAA,oC;MAAA,gC;MAAA,uC;QAIbqB,Q;QAHjB,IAzwXO,qBA AQ,CAywXf,C;UACI,OAAO,I;QACX,kBAaKB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cA Ac,oBAAU,wBAAV,EAAuB,sBAAK,KAAL,EAAvB,E;;QAEIB,OAAO,W;O;KApBX,C;4FAuBA,yB;MAAA,8D; MAAA,4F;MAAA,uC;QAe6B,UAE0,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA 8B,+BAA9B,C;QACrB,kBAAqB,UAAI,YAAJ,EAAI,oBAAJ,O;QACrB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UA AU,UAAI,cAAJ,EAAI,sBAAJ,SAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAnBX,C;8FAsBA,yB;MAAA,8 D;MAAA,4F;MAAA,uC;QAe0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mC



8D;MAAA,4F;MAAA,oC;MAAA,gC;MAAA,uC;QAE0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UA Ae,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAakB,UAAI,YAAJ,EAAI,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB ,C;UACI,cAAc,oBAAU,KAAV,EAAiB,sBAAI,KAAJ,EAAjB,EAA6B,wBAA7B,E;UACd,qB;;QAEJ,OAAO,W;O ;KApBX,C;sHAuBA,yB;MAAA,8D;MAAA,uC;QAE6B,Q;QAFzB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UA Ae,OAAO,I;QACtB,kBAAqB,UAAI,YAAJ,EAAI,oBAAJ,O;QACrB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAA U,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C;wHAuBA,y B;MAAA,8D;MAAA,uC;QAE0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kB AakB,UAAI,YAAJ,EAAI,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI, KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C;wHAuBA,yB;MAAA,8D;MAAA,uC ;QAE0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,UAAI,YAAJ,EAA I,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6B,W AA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C;wHAuBA,yB;MAAA,8D;MAAA,uC;QAE0B,Q;QAFtB,YAAY ,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,UAAI,YAAJ,EAAI,oBAAJ,O;QACIB,OAA O,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAE J,OAAO,W;O;KApBX,C;wHAuBA,yB;MAAA,8D;MAAA,uC;QAE0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,QAAQ, CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,UAAI,YAAJ,EAAI,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI, cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C; wHAuBA,yB;MAAA,8D;MAAA,uC;QAE0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I; QACtB,kBAakB,UAAI,YAAJ,EAAI,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EA AiB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C;wHAuBA,yB;MAAA,8D; MAAA,uC;QAE0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,UAAI, YAAJ,EAAI,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAj B,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C;wHAuBA,yB;MAAA,8D;MAAA,uC;QAE0B,Q; QAFtB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,UAAI,YAAJ,EAAI,oBAAJ,O;Q ACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UA Cd,qB;;QAEJ,OAAO,W;O;KApBX,C;wHAuBA,yB;MAAA,8D;MAAA,oC;MAAA,gC;MAAA,uC;QAE0B,Q;QA FtB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,UAAI,YAAJ,EAAI,oBAAJ,O;QA CIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,oBAAU,KAAV,EAAiB,sBAAI,KAAJ,EAAjB,EAA6B,wBAA7B,E;UA Cd,qB;;QAEJ,OAAO,W;O;KApBX,C;wGAuBA,yB;MAAA,8D;MAAA,uC;QAgB6B,UAE0,M;QAJhC,YAAY,w B;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAAqB,UAAI,YAAJ,EAAI,oBAAJ,O;QACrB,OAAO,S AAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBAAJ,SAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O; KApBX,C;0GAuBA,yB;MAAA,8D;MAAA,uC;QAgB0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAA Z,C;UAAe,OAAO,I;QACtB,kBAakB,UAAI,YAAJ,EAAI,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAA c,UAAU,UAAI,cAAJ,EAAI,sBAAJ,SAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KApBX,C;0GAuBA,yB;M AAA,8D;MAAA,uC;QAgB0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACt B,kBAakB,UAAI,YAAJ,EAAI,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EA AI,sBAAJ,SAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KApBX,C;0GAuBA,yB;MAAA,8D;MAAA,uC;QAg B0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,UAAI,YAAJ, EAAI,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBAAJ,SAAV,EAAwB, WAAxB,C;;QAEIB,OAAO,W;O;KApBX,C;0GAuBA,yB;MAAA,8D;MAAA,uC;QAgB0B,UAEU,M;QAJhC,YA AY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,UAAI,YAAJ,EAAI,oBAAJ,O;QACIB,O AAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBAAJ,SAAV,EAAwB,WAAxB,C;;QAEIB,OAA O,W;O;KApBX,C;0GAuBA,yB;MAAA,8D;MAAA,uC;QAgB0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAA Q,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,UAAI,YAAJ,EAAI,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UA CI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBAAJ,SAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KApBX,C;0GAuBA ,yB;MAAA,8D;MAAA,uC;QAgB0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I; QACtB,kBAakB,UAAI,YAAJ,EAAI,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAA J,EAAI,sBAAJ,SAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KApBX,C;0GAuBA,yB;MAAA,8D;MAAA,uC;

QAgB0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAAkB,UAAI,YAAJ,EAAl,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAl,sBAAJ,SAAV,EA AwB,WAAxB,C;;QAEIB,OAAO,W;O;KApBX,C;0GAuBA,yB;MAAA,8D;MAAA,oC;MAAA,gC;MAAA,uC;QA gB0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAAkB,UAAI,YAAJ, EAAl,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,oBAAU,sBAAI,cAAJ,EAAl,sBAAJ,UAAV,EA AwB,wBAAXB,E;;QAEIB,OAAO,W;O;KApBX,C;4FAuBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAgBoB,Q;QAH hB,IAIpZO,qBAAQ,CAkpZf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;Q AA+B,8B;QAA5C,akBtyoBO,W;QIBuyoBP,kBAAkB,O;QACIB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M ;UACI,cAAc,UAAU,WAAV,EA AuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KApBX,C;8 FAuBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IAIqZO,qBAAQ,CAkqZf,C;UAAe,OAAO,OA AO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB9zoBO,W;QIB+zoBP,kBA AkB,O;QACIB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EA AuB,OAAvB,C; UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;Q AiBoB,Q;QAHhB,IAIrZO,qBAAQ,CAkrZf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CA AP,IAAb,C;QAA+B,8B;QAA5C,akBt1oBO,W;QIBu1oBP,kBAAkB,O;QACIB,wBAAgB,SAAhB,gB;UAAgB,cA AA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EA AuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M; O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IAIsZO,qBAAQ,CAksZf,C;UA Ae,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB92oBO,W;Q IB+2oBP,kBAAkB,O;QACIB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EA Au B,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE; MAAA,gD;QAIBoB,Q;QAHhB,IAItZO,qBAAQ,CAktZf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa, mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBt4oBO,W;QIBu4oBP,kBAAkB,O;QACIB,wBAAgB,SAAhB,gB; UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EA AuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAE X,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IALuZO,qBAAQ,C AkuZf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,ak B95oBO,W;QIB+5oBP,kBAAkB,O;QACIB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU, WAAV,EA AuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA, gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IALvZO,qBAAQ,CAkvZf,C;UAAe,OAAO,OAAO,OAAP,C;QACc, kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBt7oBO,W;QIBu7oBP,kBAAkB,O;QACIB,wBA AgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EA AuB,OAAvB,C;UACd,MAAO,WAAI, WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IA IwZO,qBAAQ,CAkwZf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+ B,8B;QAA5C,akB98oBO,W;QIB+8oBP,kBAAkB,O;QACIB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UA CI,cAAc,UAAU,WAAV,EA AuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FA wBA,yB;MAAA,gD;MAAA,gE;MAAA,oC;MAAA,gC;MAAA,gD;QAIBoB,Q;QAHhB,IALxZO,qBAAQ,CAkxZf, C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBt+oBO ,W;QIBu+oBP,kBAAkB,O;QACIB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,cAAc,UAAU ,WAAV,EA AuB,oBAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;0GAwBA,yB;MAA A,gD;MAAA,gE;MAAA,gD;QAcI,IA12ZO,qBAAQ,CA02Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,e AAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB9/oBO,W;QIB+/oBP,kBAAkB,O;QACIB,wD;UACI,cAAc, UAAU,KAAV,EA AiB,WAAjB,EA A8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO, M;O;KArBX,C;4GAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAEI,IA33ZO,qBAAQ,CA23Zf,C;UAAe,OAAO, OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBvhpBO,W;QIBwhpBP,k BAAkB,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EA AiB,WAAjB,EA A8B,UAAK,KAAL,CAA9B,C;UACd,MA AO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAEI,IA54Z O,qBAAQ,CA44Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B ;QAA5C,akBhjpBO,W;QIBhjpBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EA AiB,WAAjB,EA A8B,U AAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;4GAyBA,yB;MAAA,gD;

MAAA,gE;MAAA,gD;QAeI,IA75ZO,qBAAQ,CA65Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBzkbBO,W;QlB0kpBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UA AU,KA AV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M; O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IA96ZO,qBAAQ,CA86Zf,C;UAAe,OAAO,OA AO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBlmpBO,W;QlBmmpBP,kB AAkB,O;QACIB,wD;UACI,cAAc,UAAU,KA AV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAA O,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IA7ZO, qBAAQ,CA+7Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;Q AA5C,akB3npBO,W;QlB4npBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UAAU,KA AV,EAAiB,WAAjB,EAA8B,U AAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD; MAAA,gE;MAAA,gD;QAeI,IAh9ZO,qBAAQ,Cag9Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa, mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBpppBO,W;QlBqppBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UA AU,KA AV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M; O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAj+ZO,qBAAQ,CAi+Zf,C;UAAe,OAAO,OA AO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB7qpBO,W;QlB8qpBP,kBA AkB,O;QACIB,wD;UACI,cAAc,UAAU,KA AV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO ,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,oC;MAAA,gD;QAe I,IAI/ZO,qBAAQ,Cak/Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA +B,8B;QAA5C,akBtspBO,W;QlBuspBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UAAU,KA AV,EAAiB,WAAjB,EA A8B,sBAAK,KAAL,EAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;gGAyBA,yB;MA AA,qD;MAAA,gE;MAAA,uC;QACI,IA1kaO,qBAAQ,CA0kaf,C;UAAe,OAAO,W;QACtB,sBAaqB,UAAK,CAA L,CAArB,C;QACgC,kBAAnB,eAAa,gBAAb,C;QAA2B,sBAAI,aAAJ,C;QAAxC,akB/tpBO,W;QlBgupBP,iBAAc, CAAd,UAAsB,gBAAtB,U;UACI,gBAAc,UAAU,aAAV,EAAuB,UAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI, aAAJ,C;;QAEX,OAAO,M;O;KArBX,C;kGAwBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAWI,IAvlaO,qBAAQ, CAulaf,C;UAAe,OAAO,W;QACtB,sBAakB,UAAK,CAAL,CAAI,C;QACmC,kBAAtB,eAAgB,gBAAhB,C;QA A8B,sBAAI,aAAJ,C;QAA3C,akBpvpBO,W;QlBqvpBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI,gBAAc,UAAU,a AAV,EAAuB,UAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAIBX,C;kGAqBA,y B;MAAA,qD;MAAA,gE;MAAA,uC;QAWI,IApmaO,qBAAQ,CAomaf,C;UAAe,OAAO,W;QACtB,sBAakB,UAA K,CAAL,CAAI,C;QACoC,kBAAvB,eAAiB,gBAajB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,akBzwpBO,W;QlB0w pBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI,gBAAc,UAAU,aAAV,EAAuB,UAAK,KAAL,CAAvB,C;UACd,MA AO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAIBX,C;kGAqBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAWI,IAjna O,qBAAQ,CAinaf,C;UAAe,OAAO,W;QACtB,sBAakB,UAAK,CAAL,CAAI,C;QACkC,kBAArB,eAAe,gBAAf, C;QAA6B,sBAAI,aAAJ,C;QAA1C,akB9xpBO,W;QlB+xpBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI,gBAAc,U AAU,aAAV,EAAuB,UAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAIBX,C;kG AqBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAWI,IA9naO,qBAAQ,CA8naf,C;UAAe,OAAO,W;QACtB,sBAak B,UAAK,CAAL,CAAI,C;QACmC,kBAAtB,eAAgB,gBAAhB,C;QAA8B,sBAAI,aAAJ,C;QAA3C,akBnzpBO,W ;QlBozpBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI,gBAAc,UAAU,aAAV,EAAuB,UAAK,KAAL,CAAvB,C;UA Cd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAIBX,C;kGAqBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAWI ,IA3oaO,qBAAQ,CA2oaf,C;UAAe,OAAO,W;QACtB,sBAakB,UAAK,CAAL,CAAI,C;QACoC,kBAAvB,eAAi B,gBAajB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,akBx0pBO,W;QlBy0pBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI ,gBAAc,UAAU,aAAV,EAAuB,UAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAI BX,C;kGAqBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAWI,IAxpaO,qBAAQ,CAwpaf,C;UAAe,OAAO,W;QACt B,sBAakB,UAAK,CAAL,CAAI,C;QACqC,kBAAxB,eAAkB,gBAAlB,C;QAAgC,sBAAI,aAAJ,C;QAA7C,akB7 1pBO,W;QlB81pBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI,gBAAc,UAAU,aAAV,EAAuB,UAAK,KAAL,CAA vB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAIBX,C;kGAqBA,yB;MAAA,qD;MAAA,gE;MAAA,u C;QAWI,IArqaO,qBAAQ,CAqqaf,C;UAAe,OAAO,W;QACtB,sBAakB,UAAK,CAAL,CAAI,C;QACsC,kBAAz B,eAAmB,gBAAnB,C;QAAiC,sBAAI,aAAJ,C;QAA9C,akBl3pBO,W;QlBm3pBP,iBAAc,CAAd,UAAsB,gBAAtB ,U;UACI,gBAAc,UAAU,aAAV,EAAuB,UAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,

M;O;KAIBX,C;kGAqBA,yB;MAAA,qD;MAAA,gE;MAAA,oC;MAAA,gC;MAAA,uC;QAWI,IAIraO,qBAAQ,CA  
kraf,C;UAAe,OAAO,W;QACtB,sBAakB,UAAK,CAAL,CAAIB,C;QACmC,kBAAtB,eAAgB,gBAAhB,C;QAA8  
B,sBAAI,0BAAJ,C;QAA3C,akBv4pBO,W;QIBw4pBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,oBAAU,  
0BAAV,EAAuB,sBAAK,KAAL,EAAvB,E;UACd,MAAO,WAAI,0BAAJ,C;;QAEX,OAAO,M;O;KAIBX,C;8GAq  
BA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QACl,IA1waO,qBAAQ,CA0waf,C;UAAe,OAAO,W;QACtB,sBAAqB,  
UAAK,CAAL,CAArB,C;QACgC,kBAAnB,eAAa,gBAAb,C;QAA2B,sBAAI,aAAJ,C;QAAXC,akB/5pBO,W;QIBg  
6pBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CA  
A9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KArBX,C;gHawBA,yB;MAAA,qD;MAAA,gE;MAA  
A,uC;QAYI,IAxxaO,qBAAQ,CAwxaf,C;UAAe,OAAO,W;QACtB,sBAakB,UAAK,CAAL,CAAIB,C;QACmC,kB  
AAtB,eAAgB,gBAAhB,C;QAA8B,sBAAI,aAAJ,C;QAA3C,akBr7pBO,W;QIBs7pBP,iBAAc,CAAd,UAAAsB,gB  
AAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAA  
J,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAYI,IAtyaO,qBAAQ,CA  
sya f,C;UAAe,OAAO,W;QACtB,sBAakB,UAAK,CAAL,CAAIB,C;QACoC,kBAAvB,eAAiB,gBAAjB,C;QAA+B,sB  
AAI,aAAJ,C;QAA5C,akB38pBO,W;QIB48pBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAAV,  
EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;g  
HAsBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAYI,IAPzaO,qBAAQ,CAozaf,C;UAAe,OAAO,W;QACtB,sBAak  
B,UAAK,CAAL,CAAIB,C;QACkC,kBAArB,eAAe,gBAaf,C;QAA6B,sBAAI,aAAJ,C;QAA1C,akBj+pBO,W;QIB  
k+pBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,C  
AA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MA  
AA,uC;QAYI,IAI0aO,qBAAQ,CAk0af,C;UAAe,OAAO,W;QACtB,sBAakB,UAAK,CAAL,CAAIB,C;QACmC,k  
BAAtB,eAAgB,gBAAhB,C;QAA8B,sBAAI,aAAJ,C;QAA3C,akBv/pBO,W;QIBw/pBP,iBAAc,CAAd,UAAAsB,gB  
AAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aA  
AJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAYI,IAh1aO,qBAAQ,CA  
glaf,C;UAAe,OAAO,W;QACtB,sBAakB,UAAK,CAAL,CAAIB,C;QACoC,kBAAvB,eAAiB,gBAAjB,C;QAA+B  
,sBAAI,aAAJ,C;QAA5C,akB7gqBO,W;QIB8gqBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAA  
V,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,  
C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAYI,IA91aO,qBAAQ,CA81af,C;UAAe,OAAO,W;QACtB,sB  
AAkB,UAAK,CAAL,CAAIB,C;QACqC,kBAAxB,eAAkB,gBAAlB,C;QAAgC,sBAAI,aAAJ,C;QAA7C,akBniqB  
O,W;QIBoiqBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,K  
AAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,g  
E;MAAA,uC;QAYI,IA52aO,qBAAQ,CA42af,C;UAAe,OAAO,W;QACtB,sBAakB,UAAK,CAAL,CAAIB,C;QAC  
sC,kBAAzB,eAAmB,gBAAnB,C;QAAiC,sBAAI,aAAJ,C;QAA9C,akBzjqBO,W;QIB0jqBP,iBAAc,CAAd,UAAAsB  
,gBAAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI  
,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,oC;MAAA,gC;MAAA,uC;Q  
AYI,IA13aO,qBAAQ,CA03af,C;UAAe,OAAO,W;QACtB,sBAakB,UAAK,CAAL,CAAIB,C;QACmC,kBAAtB,e  
AAgB,gBAAhB,C;QAA8B,sBAAI,0BAAJ,C;QAA3C,akB/kqBO,W;QIBglqBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;  
UACI,gBAAc,oBAAU,KAAV,EAAiB,0BAAjB,EAA8B,sBAAK,KAAL,EAA9B,E;UACd,MAAO,WAAI,0BAAJ,  
C;;QAEX,OAAO,M;O;KAnBX,C;8EAsBA,yB;MA+zBA,gD;MAAA,gE;MA+zBA,gD;QAcW,sB;;UA7zBS,Q;UA  
HhB,IAIpZO,qBAAQ,CAkpZf,C;YAAe,qBAAO,OA0BH,OA0BG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBA  
AO,CAAP,IAAb,C;UAA+B,sBA+zBzB,OA+zByB,C;UAA5C,akBtyoBO,W;UIBuyoBP,kBA8zBmB,O;UA7zBnB,i  
D;YAAgB,cAAhB,e;YACI,cA4zBwB,SA5zBV,CAAU,WAAV,EAAuB,OAavB,C;YACd,MAAO,WAAI,WAAJ,  
C;;UAEX,qBAAO,M;;QAYzBP,yB;O;KADJ,C;gFAiBA,yB;MAzzBA,gD;MAAA,gE;MAyzBA,gD;QAEw,sB;;UA  
vzBS,Q;UAHhB,IAIqZO,qBAAQ,CAkqZf,C;YAAe,qBAAO,OA0zBH,OA1zBG,C;YAAP,uB;;UACqB,kBAAvB,e  
AAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAyzBzB,OAzzByB,C;UAA5C,akB9zoBO,W;UIB+zobP,kBAwzBmB,O;  
UAvzBnB,iD;YAAgB,cAAhB,e;YACI,cAszBwB,SAtzBV,CAAU,WAAV,EAAuB,OAavB,C;YACd,MAAO,WAAI,WAAJ,  
C;;UAEX,qBAAO,M;;QAmzBP,yB;O;KAFJ,C;gFAkBA,yB;MANzBA,gD;MAAA,gE;MANzBA,gD;QAEw,sB;;UA  
jzBS,Q;UAHhB,IAIrZO,qBAAQ,CAkrZf,C;YAAe,qBAAO,OAozBH,OApozBG,C;YAAP,uB;;UACq  
B,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAAmzBzB,OAanzByB,C;UAA5C,akBt1oBO,W;UIBu1oBP,k

BAkzBmB,O;UAjzBnB,iD;YAAgB,cAAhB,e;YACl,cAgzBwB,SAhzBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QA6yBP,yB;O;KafJ,C;gFAkBA,yB;MA7yBA,gD;MAAA,gE;MA6yBA,gD;QAeW,sB;;UA3yBS,Q;UAHhB,IAIsZO,qBAAQ,CAksZf,C;YAAe,qBAAO,OA8yBH,OA9yBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA6yBzB,OA7yByB,C;UAA5C,akB92oBO,W;UIB+2oBP,kBA4yBmB,O;UA3yBnB,iD;YAAgB,cAAhB,e;YACl,cA0yBwB,SA1yBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAuyBP,yB;O;KafJ,C;gFAkBA,yB;MAvyBA,gD;MAAA,gE;MAuyBA,gD;QAeW,sB;;UAryBS,Q;UAHhB,IAItZO,qBAAQ,CAktZf,C;YAAe,qBAAO,OAwyBH,OAxyBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAuyBzB,OAvyByB,C;UAA5C,akBt4oBO,W;UIBu4oBP,kBA5yBmB,O;UAryBnB,iD;YAAgB,cAAhB,e;YACl,cAoyBwB,SApyBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAiyBP,yB;O;KafJ,C;gFAkBA,yB;MAjyBA,gD;MAAA,gE;MAiyBA,gD;QAeW,sB;;UA/xBS,Q;UAHhB,IALuZO,qBAAQ,CAkuZf,C;YAAe,qBAAO,OAkyBH,OAlyBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAiyBzB,OAjyByB,C;UAA5C,akB95oBO,W;UIB+5oBP,kBAgyBmB,O;UA/xBnB,iD;YAAgB,cAAhB,e;YACl,cA8xBwB,SA9xBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QA2xBP,yB;O;KafJ,C;gFAkBA,yB;MA3xBA,gD;MAAA,gE;MA2xBA,gD;QAeW,sB;;UAzxBS,Q;UAHhB,IALvZO,qBAAQ,CAkvZf,C;YAAe,qBAAO,OA4xBH,OA5xBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA2xBzB,OA3xByB,C;UAA5C,akBt7oBO,W;UIBu7oBP,kBA0xBmB,O;UAzxBnB,iD;YAAgB,cAAhB,e;YACl,cAwxBwB,SAxxBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAqxBP,yB;O;KafJ,C;gFAkBA,yB;MARxBA,gD;MAAA,gE;MAqxBA,gD;QAeW,sB;;UANxBS,Q;UAHhB,IALwZO,qBAAQ,CAkwZf,C;YAAe,qBAAO,OA4xBH,OA5xBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAqxBzB,OArxByB,C;UAA5C,akB98oBO,W;UIB+8oBP,kBAoxBmB,O;UANxBnB,iD;YAAgB,cAAhB,e;YACl,cAkxBwB,SAIxBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QA+wBP,yB;O;KafJ,C;gFAkBA,yB;MA/wBA,gD;MAAA,gE;MAAA,oC;MAAA,gC;MA+wBA,gD;QAeW,sB;;UA7wBS,Q;UAHhB,IALxZO,qBAAQ,CAkxZf,C;YAAe,qBAAO,OAgxBH,OAhxBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA+wBzB,OA/wByB,C;UAA5C,akBt+oBO,W;UIBu+oBP,kBA8wBmB,O;UA7wBnB,iD;YAAgB,cAAhB,0B;YACl,cA4wBwB,SA5wBV,CAAU,WAAV,EAAuB,oBAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAYwBP,yB;O;KafJ,C;4FAkBA,yB;MAzwBA,gD;MAAA,gE;MAYwBA,gD;QAeW,6B;;UA1wBP,IA12ZO,qBAAQ,CA02Zf,C;YAAe,4BAAO,OA0wBI,OA1wBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAywBIB,OAzwBkB,C;UAA5C,akB9/oBO,W;UIB+/oBP,kBAwwB0B,O;UAvwB1B,wD;YACl,cAswB+B,SAtwBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAmwBP,gC;O;KafJ,C;8FAkBA,yB;MANwBA,gD;MAAA,gE;MAMwBA,gD;QAgBW,6B;;UApwBP,IA33ZO,qBAAQ,CA23Zf,C;YAAe,4BAAO,OAowBI,OApwBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAmwBIB,OAAnwBkB,C;UAA5C,akBvhpBO,W;UIBwhpBP,kBAkwB0B,O;UAjwB1B,wD;YACl,cAgwB+B,SAhwBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QA6vBP,gC;O;KAhBJ,C;8FAMBA,yB;MA7vBA,gD;MAAA,gE;MA6vBA,gD;QAgBW,6B;;UA9vBP,IA54ZO,qBAAQ,CA44Zf,C;YAAe,4BAAO,OA8vBI,OA9vBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA6vBIB,OA7vBkB,C;UAA5C,akBhjpBO,W;UIBijpBP,kBA4vB0B,O;UA3vB1B,wD;YACl,cA0vB+B,SA1vBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAuvBP,gC;O;KAhBJ,C;8FAMBA,yB;MAvvBA,gD;MAAA,gE;MAuvBA,gD;QAgBW,6B;;UAXvBP,IA75ZO,qBAAQ,CA65Zf,C;YAAe,4BAAO,OAwwBI,OAxxvBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAuvBIB,OAvvBkB,C;UAA5C,akBzkpBO,W;UIB0kpBP,kBA5vB0B,O;UARvB1B,wD;YACl,cAovB+B,SApvBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAivBP,gC;O;KAhBJ,C;8FAMBA,yB;MAjvBA,gD;MAAA,gE;MAivBA,gD;QAgBW,6B;;UALvBP,IA96ZO,qBAAQ,CA86Zf,C;YAAe,4BAAO,OAkvBI,OA1vBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAivBIB,OAjvBkB,C;UAA5C,akBlmpBO,W;UIBmmpBP,kBAgvB0B,O;UA/uB1B,wD;YACl,cA8uB+B,SA9uBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QA2uBP,gC;O;KAhBJ,C;8FAMBA,yB;MA3uBA,gD;MAAA,gE;MA2uBA,gD;QAgBW,

6B;;UA5uBP,IA/7ZO,qBAAQ,CA+7Zf,C;YAAe,4BAAO,OA4uBI,OA5uBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa ,mBAAO,CAAP,IAAb,C;UAA+B,sBA2uBIB,OA3uBkB,C;UAA5C,akB3npBO,W;UIB4npBP,kBA0uB0B,O;UAz uB1B,wD;YACI,cAwuB+B,SAxuBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,M AAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAquBP,gC;O;KAhBJ,C;8FAMBA,yB;MAruBA,gD;MAAA,gE;MAq uBA,gD;QAgBW,6B;;UAtuBP,IAh9ZO,qBAAQ,CAg9Zf,C;YAAe,4BAAO,OA5uBI,OA4uBJ,C;YAAP,8B;;UACq B,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAquBIB,OAruBkB,C;UAA5C,akBpppBO,W;UIBqppBP,kB AouB0B,O;UAnuB1B,wD;YACI,cAkuB+B,SAluBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA 9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QA+tBP,gC;O;KAhBJ,C;8FAMBA,yB;MA/tBA,gD;M AAA,gE;MA+tBA,gD;QAgBW,6B;;UAhuBP,IAj+ZO,qBAAQ,CAi+Zf,C;YAAe,4BAAO,OAguBI,OAhuBJ,C;YA AP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA+tBIB,OA/tBkB,C;UAA5C,akB7qpBO,W; UIB8qpBP,kBA8tB0B,O;UA7tB1B,wD;YACI,cA4tB+B,SA5tBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK, KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAytBP,gC;O;KAhBJ,C;8FAMBA,yB;M AztBA,gD;MAAA,gE;MAAA,oC;MAytBA,gD;QAgBW,6B;;UA1tBP,IAI/ZO,qBAAQ,CAk/Zf,C;YAAe,4BAAO, OA0tBI,OA1tBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAytBIB,OAztBkB,C; UAA5C,akBtspBO,W;UIBuspBP,kBAwtB0B,O;UAvtB1B,wD;YACI,cAstB+B,SAttBjB,CAAU,KAAV,EAAiB,W AAjB,EAA8B,sBAK,KAAL,EAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAMtBP,gC;O;KA hBJ,C;gFAMBA,+B;MAOoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI, YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;kFAGX,+B;MAOoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SA AhB,gB;QAAGB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;kFAGX,+B;MAOoB ,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I ;MAEJ,OAAO,G;K;kFAGX,+B;MAOoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAh B,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;kFAGX,+B;MAOoB,Q;MADhB,UAAe,C;MACf,w BAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;kFAGX, +B;MAOoB,Q;MADhB,UAAe,C;MACf,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,YAAO,SAAS,O AAT,CAAP,I;;MAEJ,OAAO,G;K;kFAGX,+B;MAOoB,Q;MADhB,UAAe,C;MACf,wBAAGB,SAAhB,gB;QAAGB ,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;kFAGX,+B;MAOoB,Q;MADhB,UA Ae,C;MACf,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO ,G;K;kFAGX,yB;MAAA,oC;MAAA,gC;MAAA,sC;QA0oB,Q;QADhB,UAAe,C;QACf,wBAAGB,SAAhB,gB;UA AgB,cAAhB,UAGB,SAAhB,O;UACI,YAAO,SAAS,oBAAT,CAAP,I;;QAEJ,OAAO,G;O;KAVX,C;4FAaA,+B; MAOoB,Q;MADhB,UAAkB,G;MACIB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OA AT,C;;MAEX,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;MACIB,wBAAGB,SAAhB,gB;QAAGB,cA AA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;M ACIB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;8FAG X,+B;MAOoB,Q;MADhB,UAAkB,G;MACIB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SA AS,OAAT,C;;MAEX,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;MACIB,wBAAGB,SAAhB,gB;QA AgB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAA kB,G;MACIB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G; K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;MACIB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,O AAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;MACIB,wBAAGB,SAAhB ,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;8FAGX,yB;MAAA,oC;MAAA ,gC;MAAA,sC;QA0oB,Q;QADhB,UAAkB,G;QACIB,wBAAGB,SAAhB,gB;UAGB,cAAhB,UAGB,SAAhB,O; UACI,OAAO,SAAS,oBAAT,C;;QAEJ,OAAO,G;O;KAVX,C;gFAaA,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB ,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;kFAGX,+B; MAUoB,Q;MADhB,UAAoB,C;MACpB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,O AAT,C;;MAEX,OAAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB,wBAAGB,SAAhB,gB;QAAGB,c AAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAoB,C; MACpB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;kF AGX,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,OAAO,



SAAS,OAAT,C;;MAEX,OAAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB,wBAAgB,SAAhB,gB; QAAgB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,U AAoB,C;MACpB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO ,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI, OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;kFAGX,yB;MAAA,oC;MAAA,gC;MAAA,sC;QAUoB,Q;QADhB,U AAoB,C;QACpB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,OAAO,SAAS,oBAAT,C;;QAE X,OAAO,G;O;KAbX,C;kFAGBA,+B;MAUoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,S AAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAe,C;MA C f,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mF AGX,+B;MAUoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,SA AS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAUoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;Q AA gB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAUoB,Q;MADh B,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,O AAO,G;K;mFAGX,+B;MAUoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QA CI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAUoB,Q;MADhB,UAAe,C;MACf,wBAAgB, SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MA UoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CA AP,I;;MAEJ,OAAO,G;K;mFAGX,yB;MAAA,oC;MAAA,gC;MAAA,sC;QAUoB,Q;QADhB,UAAe,C;QACf,wBA AgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,YAAO,SAAS,oBAAT,CAAP,I;;QAEJ,OAAO,G;O;K AbX,C;mFAGBA,yB;MAAA,SASoB,gB;MATpB,sC;QAUoB,Q;QADhB,Y;QACA,wBAAgB,SAAhB,gB;UAAgB, cAAA,SAAhB,M;UACI,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;mFAGBA,yB;MAAA,SASoB ,gB;MATpB,sC;QAUoB,Q;QADhB,Y;QACA,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAO,SAA S,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;mFAGBA,yB;MAAA,SASoB,gB;MATpB,sC;QAUoB,Q;QADhB, Y;QACA,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G; O;KAbX,C;mFAGBA,yB;MAAA,SASoB,gB;MATpB,sC;QAUoB,Q;QADhB,Y;QACA,wBAAgB,SAAhB,gB;UA AgB,cAAA,SAAhB,M;UACI,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;mFAGBA,yB;MAAA,S ASoB,gB;MATpB,sC;QAUoB,Q;QADhB,Y;QACA,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAA O,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;mFAGBA,yB;MAAA,SASoB,gB;MATpB,sC;QAUoB,Q;Q ADhB,Y;QACA,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OA AO,G;O;KAbX,C;mFAGBA,yB;MAAA,SASoB,gB;MATpB,sC;QAUoB,Q;QADhB,Y;QACA,wBAAgB,SAAhB,g B;UAAgB,cAAA,SAAhB,M;UACI,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;mFAGBA,yB;MA AA,SASoB,gB;MATpB,sC;QAUoB,Q;QADhB,Y;QACA,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI, cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;mFAGBA,yB;MAAA,SASoB,gB;MATpB,oC;MAAA, gC;MAAA,sC;QAUoB,Q;QADhB,Y;QACA,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,cA AO,SAAS,oBAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;mFAGBA,yB;MGvwrBA,6B;MHuwrBA,sC;QAWoB,Q; QADhB,UGvwrBmC,cHuwrBnB,CGvwrBmB,C;QHwwrBnB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;U ACI,MG3ksBiD,cH2ksBjD,GG3ksB2D,KAAK,GH2ksBzD,SAAS,OAAT,CG3ksBoE,KAAX,IAAf,C;;QH6ksBrD, OAAO,G;O;KAdX,C;mFAiBA,yB;MGxxrBA,6B;MHwxrBA,sC;QAWoB,Q;QADhB,UGxxrBmC,cHwxrBnB,CG xxrBmB,C;QHyyrBnB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MG5lsBiD,cH4lsBjD,GG5lsB2D,K AAK,GH4lsBzD,SAAS,OAAT,CG5lsBoE,KAAX,IAAf,C;;QH8lsBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MGzyr BA,6B;MHyyrBA,sC;QAWoB,Q;QADhB,UGzyrBmC,cHyyrBnB,CGzyrBmB,C;QH0yrBnB,wBAAgB,SAAhB,gB ;UAAgB,cAAA,SAAhB,M;UACI,MG7msBiD,cH6msBjD,GG7msB2D,KAAK,GH6msBzD,SAAS,OAAT,CG7ms BoE,KAAX,IAAf,C;;QH+msBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MG1zrBA,6B;MH0zrBA,sC;QAWoB,Q;QA DhB,UG1zrBmC,cH0zrBnB,CG1zrBmB,C;QH2zrBnB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,M G9nsBiD,cH8nsBjD,GG9nsB2D,KAAK,GH8nsBzD,SAAS,OAAT,CG9nsBoE,KAAX,IAAf,C;;QHgosBrD,OAAO ,G;O;KAdX,C;mFAiBA,yB;MG30rBA,6B;MH20rBA,sC;QAWoB,Q;QADhB,UG30rBmC,cH20rBnB,CG30rBmB, C;QH40rBnB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MG/osBiD,cH+osBjD,GG/osB2D,KAAK,G H+osBzD,SAAS,OAAT,CG/osBoE,KAAX,IAAf,C;;QHipsBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MG51rBA,6B;

MH41rBA,sC;QAWoB,Q;QADhB,UG51rBmC,cH41rBnB,CG51rBmB,C;QH61rBnB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MGhqsBiD,cHhqsBjD,GGhqsB2D,KAAK,GHhqsBzD,SAAS,OAAT,CGhqsBoE,KAAAX,IAAf,C;;QHkqsBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MG72rBA,6B;MH62rBA,sC;QAWoB,Q;QADhB,UG72rBmC,cH62rBnB,CG72rBmB,C;QH82rBnB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MGjrsBiD,cHirsBjD,GGjrsB2D,KAAK,GHirsBzD,SAAS,OAAT,CGjrsBoE,KAAAX,IAAf,C;;QHmrsBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MG93rBA,6B;MH83rBA,sC;QAWoB,Q;QADhB,UG93rBmC,cH83rBnB,CG93rBmB,C;QH+3rBnB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MGlssBiD,cHkssBjD,GGlssB2D,KAAK,GHkssBzD,SAAS,OAAT,CGlssBoE,KAAAX,IAAf,C;;QHossBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MAAA,oC;MAAA,gC;MG/4rBA,6B;MH+4rBA,sC;QAWoB,Q;QADhB,UG/4rBmC,cH+4rBnB,CG/4rBmB,C;QHg5rBnB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,MGntsBiD,cHmntsBjD,GGntsB2D,KAAK,GHmntsBzD,SAAS,oBAAT,CgntsBoE,KAAAX,IAAf,C;;QHqtsBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MoB75rBA,+B;MpB65rBA,sC;QAWoB,Q;QADhB,UoB55rBqC,eAAW,oBpB45rB/B,CoB55rB+B,CAAX,C;QpB65rBrC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MoBjusBmD,epBiusBnD,GoBjusB8D,KAAK,KpBiusB5D,SAAS,OAAT,CoBjusBuE,KAAAX,CAAhB,C;;QpBmusBvD,OAAO,G;O;KAdX,C;mFAiBA,yB;MoB96rBA,+B;MpB86rBA,sC;QAWoB,Q;QADhB,UoB76rBqC,eAAW,oBpB66rB/B,CoB76rB+B,CAAX,C;QpB86rBrC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MoBlvsBmD,epBkvsBnD,GoBlvsB8D,KAAK,KpBkvsB5D,SAAS,OAAT,CoBlvsBuE,KAAAX,CAAhB,C;;QpBovsBvD,OAAO,G;O;KAdX,C;mFAiBA,yB;MoB/7rBA,+B;MpB+7rBA,sC;QAWoB,Q;QADhB,UoB97rBqC,eAAW,oBpB87rB/B,CoB97rB+B,CAAX,C;QpB+7rBrC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MoBnwsBmD,epBmwsBnD,GoBnwsB8D,KAAK,KpBmwsB5D,SAAS,OAAT,CoBnwsBuE,KAAAX,CAAhB,C;;QpBqwsBvD,OAAO,G;O;KAdX,C;kFAiBA,yB;MoBh9rBA,+B;MpBg9rBA,sC;QAWoB,Q;QADhB,UoB/8rBqC,eAAW,oBpB+8rB/B,CoB/8rB+B,CAAX,C;QpBg9rBrC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MoBpxsBmD,epBoxsBnD,GoBpxsB8D,KAAK,KpBoxsB5D,SAAS,OAAT,CoBpxsBuE,KAAAX,CAAhB,C;;QpBsxsBvD,OAAO,G;O;KAdX,C;mFAiBA,yB;MoBj+rBA,+B;MpBi+rBA,sC;QAWoB,Q;QADhB,UoBh+rBqC,eAAW,oBpBg+rB/B,CoBh+rB+B,CAAX,C;QpBi+rBrC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MoBrysBmD,epBqysBnD,GoBrysB8D,KAAK,KpBqysB5D,SAAS,OAAT,CoBrysBuE,KAAAX,CAAhB,C;;QpBuysBvD,OAAO,G;O;KAdX,C;mFAiBA,yB;MoBl/rBA,+B;MpBk/rBA,sC;QAWoB,Q;QADhB,UoBj/rBqC,eAAW,oBpBi/rB/B,CoBj/rB+B,CAAX,C;QpBk/rBrC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MoBtzsBmD,epBszsBnD,GoBtzsB8D,KAAK,KpBszsB5D,SAAS,OAAT,CoBtzsBuE,KAAAX,CAAhB,C;;QpBwzsBvD,OAAO,G;O;KAdX,C;mFAiBA,yB;MoBngsBA,+B;MpBmgsBA,sC;QAWoB,Q;QADhB,UoBlgsBqC,eAAW,oBpBkgsB/B,CoBlgsB+B,CAAX,C;QpBmgsBrC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MoBv0sBmD,epBu0sBnD,GoBv0sB8D,KAAK,KpBu0sB5D,SAAS,OAAT,CoBv0sBuE,KAAAX,CAAhB,C;;QpBy0sBvD,OAAO,G;O;KAdX,C;kFAiBA,yB;MoBphsBA,+B;MpBohsBA,sC;QAWoB,Q;QADhB,UoBnhsBqC,eAAW,oBpBmhsB/B,CoBnhsB+B,CAAX,C;QpBohsBrC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MoBx1sBmD,epBw1sBnD,GoBx1sB8D,KAAK,KpBw1sB5D,SAAS,OAAT,CoBx1sBuE,KAAAX,CAAhB,C;;QpB01sBvD,OAAO,G;O;KAdX,C;mFAiBA,yB;MAAA,oC;MAAA,gC;MoBrisBA,+B;MpBqisBA,sC;QAWoB,Q;QADhB,UoBpisBqC,eAAW,oBpBoisB/B,CoBpisB+B,CAAX,C;QpBqisBrC,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,MoBz2sBmD,epBy2sBnD,GoBz2sB8D,KAAK,KpBy2sB5D,SAAS,oBAAT,CoBz2sBuE,KAAAX,CAAhB,C;;QpB22sBvD,OAAO,G;O;KAdX,C;IAiBA,mC;MAIoB,UAMT,M;MANP,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,eAAJ,C;UACI,MAAM,gCAAYB,2BAAwB,SAAxB,MAAZB,C;;MAId,OAAO,0D;K;wFAGX,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;QAGf,OAAO,cAAK,KAAL,EAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;QAGf,OAAO,cAAK,KAAL,EAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;QAGf,OAAO,cAAK,KAAL,EAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAgB,S



AIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAA  
O,I;O;KAbX,C;8EAgBA,yB;MAAA,gE;MmB9gtBA,iB;MnB8gtBA,8C;QAQI,WmBhhtBO,MAAO,KnBghtBG,gB  
mBhhtBH,EnBghtBS,KAAM,OmBhhtBf,C;QnBihtBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAI  
B,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,  
I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmB9htBA,iB;MnB8htBA,8C;QAQI,WmBhitBO,MAAO,KnBghtBG,gBm  
BhitBH,EnBghtBS,KAAM,OmBhitBf,C;QnBihtBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAI  
B,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;  
O;KAbX,C;+EAgBA,yB;MAAA,gE;MmB9itBA,iB;MnB8itBA,8C;QAQI,WmBhjtBO,MAAO,KnBgjtBG,gBmBhj  
tBH,EnBgjtBS,KAAM,OmBhjtBf,C;QnBijtBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAI  
B,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;K  
AbX,C;+EAgBA,yB;MAAA,gE;MmB9jtBA,iB;MnB8jtBA,8C;QAQI,WmBhktBO,MAAO,KnBgktBG,gBmBhktB  
H,EnBgktBS,KAAM,OmBhktBf,C;QnBiktBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAI  
B,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;K  
AbX,C;+EAgBA,yB;MAAA,gE;MmB9ktBA,iB;MnB8ktBA,8C;QAQI,WmBhltBO,MAAO,KnBgltBG,gBmBhltB  
H,EnBgltBS,KAAM,OmBhltBf,C;QnBiltBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAI  
B,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAb  
X,C;+EAgBA,yB;MAAA,gE;MAAA,oC;MmB9ltBA,iB;MnB8ltBA,8C;QAQI,WmBhmtBO,MAAO,KnBgmtBG,g  
BmBhmtBH,EnBgmtBS,KAAM,OmBhmtBf,C;QnBimtBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,I  
AAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,EA AV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,O  
AAO,I;O;KAbX,C;IAgBA,kC;MAqGoB,gB;MAHhB,gBAAGB,gB;MACHB,WAAW,iBmB3stBJ,MAAO,KnB2stB  
sB,wBA5FzB,KA4FyB,EAawB,EAaxB,CmB3stBtB,EnB2stBmD,SmB3stBnD,CnB2stBH,C;MACX,QAAQ,C;M  
ACQ,OA9FL,KA8FK,W;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACp  
B,IAAK,WahGqB,GAGP,UAAK,UAAL,EA AK,kBAAL,SAhGO,EAGGI,OA hGJ,CAGrB,C;;MAhGT,OAKGO,I  
;K;IA/FX,kC;MA6GoB,gB;MAHhB,gBAAGB,gB;MACHB,WAAW,iBmB7ttBJ,MAAO,KnB6ttBsB,wBApGzB,K  
AoGyB,EAawB,EAaxB,CmB7ttBtB,EnB6ttBmD,SmB7ttBnD,CnB6ttBH,C;MACX,QAAQ,C;MACQ,OAtGL,K  
AsGK,W;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WaxG  
qB,GAwGP,UAAK,UAAL,EA AK,kBAAL,SAxGO,EAwGI,OA xGJ,CAwGrB,C;;MAxGT,OA0GO,I;K;IAvGX,kC  
;MAqHoB,gB;MAHhB,gBAAGB,gB;MACHB,WAAW,iBmB/utBJ,MAAO,KnB+utBsB,wBA5GzB,KA4GyB,EA  
awB,EAaxB,CmB/utBtB,EnB+utBmD,SmB/utBnD,CnB+utBH,C;MACX,QAAQ,C;MACQ,OA9GL,KA8GK,W;M  
AAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WahHqB,GAGHP,  
UAAK,UAAL,EA AK,kBAAL,SAhHO,EAGHI,OA hHJ,CAGrB,C;;MAhHT,OAKHO,I;K;IA/GX,kC;MA6HoB,gB;  
MAHhB,gBAAGB,gB;MACHB,WAAW,iBmBjwBJ,MAAO,KnBiwtBsB,wBApHzB,KAoHyB,EAawB,EAaxB,C  
mBjwBtB,EnBiwtBmD,SmBjwBnD,CnBiwtBH,C;MACX,QAAQ,C;MACQ,OAtHL,KAsHK,W;MAAhB,OAAG  
B,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WaxHqB,GawHP,UAAK,UA  
AL,EA AK,kBAAL,SAxHO,EAwHI,OA xHJ,CAwHrB,C;;MAxHT,OA0HO,I;K;IAvHX,kC;MAqIoB,gB;MAHhB,gB  
AAAGB,gB;MACHB,WAAW,iBmBnxtBJ,MAAO,KnBmxtBsB,wBA5HzB,KA4HyB,EAawB,EAaxB,CmBnxtBtB,  
EnBmxtBmD,SmBnxtBnD,CnBmxtBH,C;MACX,QAAQ,C;MACQ,OA9HL,KA8HK,W;MAAhB,OAAGB,cAAhB,  
C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WahIqB,GAGIP,UAAK,UAAL,EA AK,kB  
AAL,SAhIO,EAGII,OA hIJ,CAGrB,C;;MAhIT,OAKIO,I;K;IA/HX,kC;MA6IoB,gB;MAHhB,gBAAGB,gB;MACHB,  
WAAW,iBmBrytBJ,MAAO,KnBqytBsB,wBApIzB,KAoIyB,EAawB,EAaxB,CmBrytBtB,EnBqytBmD,SmBrytB  
nD,CnBqytBH,C;MACX,QAAQ,C;MACQ,OAtIL,KAsIK,W;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI  
,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WaxIqB,GawIP,UAAK,UAAL,EA AK,kBAAL,SAxIO,EAwII,OA xIJ  
,CAwIrB,C;;MAxIT,OA0IO,I;K;IAvIX,kC;MAqJoB,gB;MAHhB,gBAAGB,gB;MACHB,WAAW,iBmBvztBJ,MA  
AO,KnBuztBsB,wBA5IzB,KA4IyB,EAawB,EAaxB,CmBvztBtB,EnBuztBmD,SmBvztBnD,CnBuztBH,C;MACX,  
QAAQ,C;MACQ,OA9IL,KA8IK,W;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAo  
B,K;QACpB,IAAK,WahJqB,GAGJP,UAAK,UAAL,EA AK,kBAAL,SAhJO,EAGJI,OA hJJ,CAGrB,C;;MAhJT,OA  
kJO,I;K;IA/IX,kC;MA6JoB,gB;MAHhB,gBAAGB,gB;MACHB,WAAW,iBmBz0tBJ,MAAO,KnBy0tBsB,wBApJz  
B,KAoJyB,EAawB,EAaxB,CmBz0tBtB,EnBy0tBmD,SmBz0tBnD,CnBy0tBH,C;MACX,QAAQ,C;MACQ,OAtJ

L,KAsJK,W;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WA  
xJqB,GAwJP,UAAK,UAAAL,EAAK,kBAAL,SAxJO,EAwJI,OAxJJ,CAwJrB,C;;MAxJT,OA0JO,I;K;IAvJX,kC;MA  
qKoB,gB;MAHhB,gBAAGB,gB;MACHB,WAAW,iBmB31tBJ,MAAO,KnB21tBsB,wBA5JzB,KA4JyB,EAAwB,E  
AAxB,CmB31tBtB,EnB21tBmD,SmB31tBnD,CnB21tBH,C;MACX,QAAQ,C;MACQ,OA9JL,KA8JK,W;MAAhB,  
OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAhKqB,GAgKP,sBAA  
K,UAAAL,EAAK,kBAAL,UAhKO,EAgKI,OAhKJ,CAGKrB,C;;MAhKT,OakKO,I;K;+EA/JX,yB;MAAA,kF;MAA  
A,gE;MmBxstBA,iB;MnBwstBA,8C;QAWoB,UAEY,M;QAL5B,gBAAGB,gB;QACHB,WAAW,emB3stBJ,MAAO  
,KnB2stBsB,wBAAN,KAAM,EAAwB,EAAxB,CmB3stBtB,EnB2stBmD,SmB3stBnD,CnB2stBH,C;QACX,QAA  
Q,C;QACQ,uB;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,  
WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+Eak  
BA,yB;MAAA,kF;MAAA,gE;MmB1ttBA,iB;MnB0ttBA,8C;QAWoB,UAEY,M;QAL5B,gBAAGB,gB;QACHB,W  
AAW,emB7ttBJ,MAAO,KnB6ttBsB,wBAAN,KAAM,EAAwB,EAAxB,CmB7ttBtB,EnB6ttBmD,SmB7ttBnD,CnB  
6ttBH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAo  
B,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO  
,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MmB5utBA,iB;MnB4utBA,8C;QAWoB,UAEY,M;QAL5B,gB  
AAGB,gB;QACHB,WAAW,emB/utBJ,MAAO,KnB+utBsB,wBAAN,KAAM,EAAwB,EAAxB,CmB/utBtB,EnB+utB  
mD,SmB/utBnD,CnB+utBH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,  
KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,  
CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MmB9vtBA,iB;MnB8vtBA,8C;QAWoB  
,UAEY,M;QAL5B,gBAAGB,gB;QACHB,WAAW,emBjwBJ,MAAO,KnBjwBsB,wBAAN,KAAM,EAAwB,EAAx  
B,CmBjwBtB,EnBjwBmD,SmBjwBnD,CnBjwBH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAGB,cAAhB,C;U  
AAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,S  
AAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MmBhxtBA,iB;  
MnBgxtBA,8C;QAWoB,UAEY,M;QAL5B,gBAAGB,gB;QACHB,WAAW,emBnxtBJ,MAAO,KnBmxtBsB,wBAA  
N,KAAM,EAAwB,EAAxB,CmBnxtBtB,EnBmxtBmD,SmBnxtBnD,CnBmxtBH,C;QACX,QAAQ,C;QACQ,uB;Q  
AAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UA  
AK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;  
MAAA,gE;MmBlytBA,iB;MnBklytBA,8C;QAWoB,UAEY,M;QAL5B,gBAAGB,gB;QACHB,WAAW,emBrytBJ,M  
AAO,KnBqytBsB,wBAAN,KAAM,EAAwB,EAAxB,CmBrytBtB,EnBqytBmD,SmBrytBnD,CnBqytBH,C;QACX,  
QAAQ,C;QACQ,uB;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IA  
AK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+  
EakBA,yB;MAAA,kF;MAAA,gE;MmBpztBA,iB;MnBoztBA,8C;QAWoB,UAEY,M;QAL5B,gBAAGB,gB;QACH  
B,WAAW,emBvztBJ,MAAO,KnBuztBsB,wBAAN,KAAM,EAAwB,EAAxB,CmBvztBtB,EnBuztBmD,SmBvztBn  
D,CnBuztBH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C  
;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,  
OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MmBt0tBA,iB;MnBs0tBA,8C;QAWoB,UAEY,M;QAL5  
B,gBAAGB,gB;QACHB,WAAW,emBz0tBJ,MAAO,KnBy0tBsB,wBAAN,KAAM,EAAwB,EAAxB,CmBz0tBtB,E  
nBy0tBmD,SmBz0tBnD,CnBy0tBH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ  
,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,O  
AARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MAAA,oC;MmBx1tBA,iB;MnB  
w1tBA,8C;QAWoB,UAEY,M;QAL5B,gBAAGB,gB;QACHB,WAAW,emB31tBJ,MAAO,KnB21tBsB,wBAAN,K  
AAM,EAAwB,EAAxB,CmB31tBtB,EnB21tBmD,SmB31tBnD,CnB21tBH,C;QACX,QAAQ,C;QACQ,uB;QAAhB  
,OAAGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,sBAAK,U  
AAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;IAkBA,kC;MAwFI,WmB57tB  
O,MAAO,KnB47tBG,gBmB57tBH,EnB22tBH,KAIkFB,OmB57tBf,C;MnB67tBd,WAAW,iBAAa,IAAb,C;MACX,  
aAAU,CAAV,MAAkB,IAAlB,M;QACI,IAAK,WApFqB,GAoFP,UAAK,CAAL,CAPFO,EAAAnB,KAOFPqB,CAAM  
,CAAN,CAPFF,CAoFrB,C;;MAPFT,OAsFO,I;K;IANFX,kC;MA8FI,WmB58tBO,MAAO,KnB48tBG,gBmB58tBH,  
EnBq3tBH,KAuFkB,OmB58tBf,C;MnB68tBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAlB,M;

QACI,IAAK,WA1FqB,GA0FP,UAAK,CAAL,CA1FO,EAAAnB,KA0FqB,CAAM,CAAN,CA1FF,CA0FrB,C;;MA1  
FT,OA4FO,I;K;IAzFX,kC;MAoGI,WmB59tBO,MAAO,KnB49tBG,gBmB59tBH,EnB+3tBH,KA6FkB,OmB59tBf,  
C;MnB69tBd,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAhGqB,GAgGP,  
UAAK,CAAL,CAhGO,EAAAnB,KAgGqB,CAAM,CAAN,CAhGF,CAGrB,C;;MAhGT,OAKGO,I;K;IA/FX,kC;M  
A0GI,WmB5+tBO,MAAO,KnB4+tBG,gBmB5+tBH,EnBy4tBH,KAmGkB,OmB5+tBf,C;MnB6+tBd,WAAW,iBA  
Aa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAtGqB,GAsGP,UAAK,CAAL,CAtGO,EAA  
nB,KAsGqB,CAAM,CAAN,CAtGF,CAsGrB,C;;MAtGT,OAwoGO,I;K;IArGX,kC;MAGHI,WmB5/tBO,MAAO,Kn  
B4/tBG,gBmB5/tBH,EnBm5tBH,KAyGkB,OmB5/tBf,C;MnB6/tBd,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV  
,MAAkB,IAAIB,M;QACI,IAAK,WA5GqB,GA4GP,UAAK,CAAL,CA5GO,EAAAnB,KA4GqB,CAAM,CAAN,CA  
5GF,CA4GrB,C;;MA5GT,OA8GO,I;K;IA3GX,kC;MAshI,WmB5guBO,MAAO,KnB4guBG,gBmB5guBH,EnB65t  
BH,KA+GkB,OmB5guBf,C;MnB6guBd,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,  
IAAK,WAIHqB,GakHP,UAAK,CAAL,CAIHQ,EAAAnB,KakHqB,CAAM,CAAN,CAIHf,CakHrB,C;;MAIHT,O  
AoHO,I;K;IAjHX,kC;MA4HI,WmB5huBO,MAAO,KnB4huBG,gBmB5huBH,EnBu6tBH,KAqHkB,OmB5huBf,C;  
MnB6huBd,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAxHqB,GAwHP,U  
AAK,CAAL,CAXHO,EAAAnB,KAwHqB,CAAM,CAAN,CAXHF,CAwHrB,C;;MAxHT,OA0HO,I;K;IAvHX,kC;M  
AkII,WmB5iuBO,MAAO,KnB4iuBG,gBmB5iuBH,EnBi7tBH,KA2HkB,OmB5iuBf,C;MnB6iuBd,WAAW,iBAaA,  
IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA9HqB,GA8HP,sBAAK,CAAL,EA9HO,EA8H  
E,YA9HrB,KA8HqB,CAAM,CAAN,EA9HF,CA8HrB,C;;MA9HT,OAgiO,I;K;+EA7HX,yB;MAAA,gE;MmB17tB  
A,iB;MnB07tBA,8C;QAQI,WmB57tBO,MAAO,KnB47tBG,gBmB57tBH,EnB47tBS,KAAM,OmB57tBf,C;QnB67  
tBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,C  
AAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmB18tB  
A,iB;MnB08tBA,8C;QAQI,WmB58tBO,MAAO,KnB48tBG,gBmB58tBH,EnB48tBS,KAAM,OmB58tBf,C;QnB68  
tBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,C  
AAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmB19tB  
A,iB;MnB09tBA,8C;QAQI,WmB59tBO,MAAO,KnB49tBG,gBmB59tBH,EnB49tBS,KAAM,OmB59tBf,C;QnB69  
tBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,C  
AAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmB1+tB  
A,iB;MnB0+tBA,8C;QAQI,WmB5+tBO,MAAO,KnB4+tBG,gBmB5+tBH,EnB4+tBS,KAAM,OmB5+tBf,C;QnB6  
+tBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,  
CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmB1/tB  
A,iB;MnB0/tBA,8C;QAQI,WmB5/tBO,MAAO,KnB4/tBG,gBmB5/tBH,EnB4/tBS,KAAM,OmB5/tBf,C;QnB6/tBd  
,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAA  
V,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmB1guBA,i  
B;MnB0guBA,8C;QAQI,WmB5guBO,MAAO,KnB4guBG,gBmB5guBH,EnB4guBS,KAAM,OmB5guBf,C;QnB6g  
uBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,C  
AAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmB1huB  
A,iB;MnB0huBA,8C;QAQI,WmB5huBO,MAAO,KnB4huBG,gBmB5huBH,EnB4huBS,KAAM,OmB5huBf,C;Qn  
B6huBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CA  
AL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MAA  
A,oC;MmB1iuBA,iB;MnB0iuBA,8C;QAQI,WmB5iuBO,MAAO,KnB4iuBG,gBmB5iuBH,EnB4iuBS,KAAM,Om  
B5iuBf,C;QnB6iuBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU  
,sBAAK,CAAL,EAaV,EAAMB,kBAAM,CAAN,EAAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;IAgBA,4F;MAQ8  
D,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;M  
AAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;MAGvN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,Y  
AAy,C;MACZ,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gB  
AAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAAIB,C;UACW,gBAAP,MAAO,EAAC,OAAd,EAau  
B,SAAvB,C;;UACJ,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAAIB,C;QAAiC,MAAO,gBAAO,SAAP,C;MA  
CxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQwD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,S

AAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAA  
A,YAAuC,I;MAGpN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAg  
B,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,  
IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAEP,MAAO,gB  
AAO,OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,S  
AAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQyD,yB;QAAA,YAA0B,I;MAAM,s  
B;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAA  
O,yB;QAAA,YAAwC,I;MAGtN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAgB,SAAh  
B,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,Q  
AAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAE  
P,MAAO,gBAAO,OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAA  
O,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQuD,yB;QAAA,YAA0  
B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YA  
A0B,K;MAAO,yB;QAAA,YAAcI,I;MAGIN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBA  
AgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QAC  
xB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAA  
P,C;;YAEP,MAAO,gBAAO,OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QA  
AiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQwD,yB;QA  
AA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;Q  
AAA,YAA0B,K;MAAO,yB;QAAA,YAAuC,I;MAGpN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;M  
ACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SA  
AP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,  
OAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,K  
AA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MA  
QyD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;  
MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAwC,I;MAGtN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,  
YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,g  
BAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAA  
O,UAAU,OAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,  
QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAG  
X,8F;MAQ0D,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA  
,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAyC,I;MAGxN,Q;MAFhB,MAAO,gBAAO,MAAP,  
C;MACP,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB  
,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MA  
AO,gBAAO,UAAU,OAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,C  
AAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAA  
O,M;K;IAGX,8F;MAQ2D,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAA  
I,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAA0C,I;MAG1N,Q;MAFhB,MAAO,gBA  
AO,MAAP,C;MACP,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAA  
d,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;  
YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI  
,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MA  
CP,OAAO,M;K;IAGX,8F;MAQwD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAw  
B,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC,I;MAGpN,Q;MAFhB,M  
AAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAhB,UAAgB,SAAhB,O;QAC  
I,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;U  
ACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,oBAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAP,C;;UACR,K;  
;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,

OAAP,C;MACP,OAAO,M;K;IAGX,0F;MAQyC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAawB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;MACIN,OAAO,kBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAASe,SAAtE,CAAI,F,W;K;IAG5F,4F;MAQkC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAawB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC,I;MAC9M,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAASe,SAAtE,CAAI,F,W;K;IAG5F,4F;MAQmC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAawB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAwC,I;MACHN,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAASe,SAAtE,CAAI,F,W;K;IAG5F,4F;MAQiC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAawB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAcC,I;MAC5M,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAASe,SAAtE,CAAI,F,W;K;IAG5F,4F;MAQkC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAawB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC,I;MAC9M,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAASe,SAAtE,CAAI,F,W;K;IAG5F,4F;MAQoC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAawB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAyC,I;MACIN,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAASe,SAAtE,CAAI,F,W;K;IAG5F,4F;MAQqC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAawB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAA0C,I;MACpN,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAASe,SAAtE,CAAI,F,W;K;IAG5F,4F;MAQkC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAawB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC,I;MAC9M,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAASe,SAAtE,CAAI,F,W;K;IAQxE,4C;MAAA,mB;QAAE,OAAK,qBAAL,eAAK,C;O;K;IAL3B,+B;MAII,IAh6fO,qBAAQ,CAg6ff,C;QAAe,OAAO,W;MACtB,kCAAgB,4BAAhB,C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,yBAAL,eAAK,C;O;K;IAL3B,iC;MAII,IAh6fO,qBAAQ,CAg6ff,C;QAAe,OAAO,W;MACtB,kCAAgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,0BAAL,eAAK,C;O;K;IAL3B,iC;MAII,IAh6fO,qBAAQ,CAg6ff,C;QAAe,OAAO,W;MACtB,kCAAgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,yBAAL,eAAK,C;O;K;IAL3B,iC;MAII,IAh6fO,qBAAQ,CAg6ff,C;QAAe,OAAO,W;MACtB,kCAAgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,0BAAL,eAAK,C;O;K;IAL3B,iC;MAII,IAh6fO,qBAAQ,CAg6ff,C;QAAe,OAAO,W;MACtB,kCAAgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,yBAAL,eAAK,C;O;K;IAL3B,iC;MAII,IAh6fO,qBAAQ,CAg6ff,C;QAAe,OAAO,W;MACtB,kCAAgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,0BAAL,eAAK,C;O;K;IAL3B,iC;MAII,IAh6fO,qBAAQ,CAg6ff,C;QAAe,OAAO,W;MACtB,kCAAgB,8BAAhB,C;K;IAUgB,4C;MAAA,mB;QAAE,OAAK,qBAAL,eAAK,C;O;K;IAP3B,+B;MAMI,IA1+fO,qBAAQ,CA0+ff,C;QAAe,OAAO,e;MACtB,kCAAgB,4BAAhB,C;K;IAUgB,8C;MAAA,mB;QAAE,OAAK,yBAAL,eAAK,C;O;K;IAP3B,iC;MAMI,IA5+fO,qBAAQ,CA4+ff,C;QAAe,OAAO,e;MACtB,kCAAgB,8BAAhB,C;K;IAUgB,8C;MAAA,mB;QAAE,OAAK,0BAAL,eAAK,C;O;K;IAP3B,iC;MAMI,IA9+fO,qBAAQ,CA8+ff,C;QAAe,OAAO,e;MACtB,kCAAgB,8BAAhB,C;K;IAUgB,8C;MAAA,mB;QAAE,OAAK,wBAAL,eAAK,C;O;K;IAP3B,iC;MAMI,IAh/fO,qBAAQ,CAg/ff,C;QAAe,OAAO,e;MACtB,kCAAgB,8BAAhB,C;K;IAUgB,8C;MAAA,mB;QAAE,OAAK,yBAAL,eAAK,C;O;K;IAP3B,iC;MAMI,IAI/fO,qBAAQ,CAk/ff,C;QAAe,OAAO,e;MACtB,kCAAgB,8BAAhB,C;K;IAUgB,8C;MAAA,mB;QAAE,OAAK,0BAAL,eAAK,C;O;K;IAP3B,iC;MAMI,IAp/fO,qBAAQ,CAo/ff,C;QAAe,OAAO,e;MACtB,kCAAgB,8BAAhB,C;K;IAUgB,8C;MAAA,mB;QAAE,OA



AK,2BAAL,eAAK,C;O;K;IAP3B,iC;MAMI,IAt/fo,qBAAQ,CAs/ff,C;QAAe,OAAO,e;MACtB,kCAAgB,8BAAhB ,C;K;IAUgB,8C;MAAA,mB;QAAE,OAAK,4BAAL,eAAK,C;O;K;IAP3B,iC;MAMI,IAx/fo,qBAAQ,CAw/ff,C;Q AAe,OAAO,e;MACtB,kCAAgB,8BAAhB,C;K;IAUgB,8C;MAAA,mB;QAAE,OAAK,yBAAL,eAAK,C;O;K;IAP3 B,iC;MAMI,IA1/fo,qBAAQ,CA0/ff,C;QAAe,OAAO,e;MACtB,kCAAgB,8BAAhB,C;K;IAGJ,4B;MAOoB,Q;MA FhB,UAAkB,G;MACIB,YAAiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QA CP,qB;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAOoB,Q;MAFh B,UAAkB,G;MACIB,YAAiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP, qB;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAOoB,Q;MAFhB, UAAkB,G;MACIB,YAAiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,q B;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAOoB,Q;MAFhB,U AAkB,G;MACIB,YAAiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;; MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAOoB,Q;MAFhB,UAA kB,G;MACIB,YAAiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;M AEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAOoB,Q;MAFhB,UAAk B,G;MACIB,YAAiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MAE J,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAMoB,Q;MAFhB,UAAkB, G;MACIB,YAAiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MAEJ, OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAMoB,Q;MAFhB,UAAkB,G;M ACIB,YAAiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MAEJ,OA AW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAMoB,Q;MAFhB,UAAkB,G;MA CIB,YAAiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MAEJ,OAA W,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAMoB,Q;MAFhB,UAAkB,G;MACI B,YAAiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MAEJ,OAAW, UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MAMoB,Q;MAFhB,UAAkB,G;MACIB, YAAiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MAEJ,OAAW,U AAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,wB;MAMoB,Q;MADhB,UAAe,C;MACf,wB AAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,O;;MAEX,OAAO,G;K;IAGX,0B;MAMoB,Q;MADh B,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,O;;MAEX,OAAO,G;K;IAGX,0 B;MAMoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,OAAP,I;; MAEJ,OAAO,G;K;IAGX,0B;MAMoB,Q;MADhB,Y;MACA,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QA CCl,cAAO,OAAP,C;;MAEJ,OAAO,G;K;IAGX,0B;MAMoB,Q;MADhB,UAAiB,G;MACjB,wBAAgB,SAAhB,gB; QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;;MAEX,OAAO,G;K;IAGX,0B;MAMoB,Q;MADhB,UAAkB,G;MACI B,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;;MAEX,OAAO,G;K;IAGX,0B;MAKoB,Q;M ADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,O;;MAEX,OAAO,G;K;IA GX,0B;MAKoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,O;;M AEX,OAAO,G;K;IAGX,0B;MAKoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M ;QACI,YAAO,OAAP,I;;MAEJ,OAAO,G;K;IAGX,0B;MAKoB,Q;MADhB,Y;MACA,wBAAgB,SAAhB,gB;QAAg B,cAAA,SAAhB,M;QACI,cAAO,OAAP,C;;MAEJ,OAAO,G;K;IAGX,0B;MAKoB,Q;MADhB,UAAiB,G;MACjB, wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAKoB,Q;MA DhB,UAAkB,G;MACIB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;;MAEX,OAAO,G;K;Ic jkwBX,oD;MAQuF,wC;K;IARvF,8CASI,Y;MAAuC,8B;K;IAT3C,gF;4FOOA,qB;MAOI,OAAO,sBAAI,CAAJ,C; K;4FAGX,qB;MAOI,OAAO,sBAAI,CAAJ,C;K;4FAGX,qB;MAOI,OAAO,sBAAI,CAAJ,C;K;4FAGX,qB;MAOI, OAAO,sBAAI,CAAJ,C;K;4FAGX,qB;MAOI,OAAO,sBAAI,CAAJ,C;K;IAGX,wC;MAII,IAAI,oCAAJ,C;QACI,O AAO,yBAAS,OAAT,C;MACX,OAAO,qBAAQ,OAAR,KAAoB,C;K;IAWG,yC;MAAA,qB;QAAE,MAAM,8BAA 0B,iDAA8C,aaA9C,MAA1B,C;O;K;IAR1C,qC;MAMI,IAAI,8BAAJ,C;QACI,OAAO,sBAAI,KAAJ,C;MACX,O AAO,6BAAgB,KAAhB,EAAuB,uBAAvB,C;K;0FAGX,4B;MAOI,OAAO,sBAAI,KAAJ,C;K;IAGX,2D;MAcQB,Q

;MARjB,IAAI,8BAAJ,C;QACI,OAAaB,KA8Lf,IAAS,CAAT,IA9Le,KA8LD,IAAS,iBA9LvB,SA8LuB,CAA3B,G  
A9LI,SA8LkC,aA9LnB,KA8LmB,CAAtC,GA9L0B,YA8L4B,CA9LnC,KA8LmC,C;;MA7L7D,IAAI,QAAQ,CAA  
Z,C;QACI,OAAO,aAAa,KAAb,C;MACX,eAAe,oB;MACf,YAAy,C;MACZ,OAAO,QAAS,UAAhB,C;QACI,cAA  
c,QAAS,O;QACvB,IAAI,WAAS,YAAT,EAAS,oBAAT,OAAJ,C;UACI,OAAO,O;;MAEf,OAAO,aAAa,KAAb,C;  
K;sGAGX,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAaC,sBAAI,KAJ,  
CAAtC,GAAaD,aAAa,KAAb,C;O;KAPjE,C;IAUA,6C;MAcqB,Q;MARjB,IAAI,8BAAJ,C;QACI,OAAy,YAAL,S  
AAK,EAAU,KAaV,C;MACHB,IAAI,QAAQ,CAAZ,C;QACI,OAAO,I;MACX,eAAe,oB;MACf,YAAy,C;MACZ,  
OAAO,QAAS,UAAhB,C;QACI,cAAc,QAAS,O;QACvB,IAAI,WAAS,YAAT,EAAS,oBAAT,OAAJ,C;UACI,OA  
AO,O;;MAEf,OAAO,I;K;sGAGX,yB;MAAA,sD;MAAA,mC;QAOI,OAAy,UAAL,SAAK,EAAU,KAaV,C;O;KA  
PhB,C;gFAUA,gC;MAOW,sB;;QAYHS,Q;QAAA,2B;QAaHb,OAaGb,cAAhB,C;UAaGb,yB;UAAM,IAzHH,SAy  
HO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MA1HP,yB;K;wFAGJ,gC;MA6  
VoB,Q;MADhB,WAAe,I;MACC,2B;MAaHb,OAaGb,cAAhB,C;QAaGb,yB;QACZ,IAvVc,SAuVV,CAAU,OAA  
V,CAAJ,C;UACI,OAAO,O;;;MAxVf,OA2VO,I;K;wFAxVX,gC;MAOW,qB;;QA0VP,eAAoB,+BAAa,cAAb,C;QA  
CpB,OAAO,QAAS,cAAhB,C;UACI,cAAc,QAAS,W;UACvB,IA7Vc,SA6VV,CAAU,OAAV,CAAJ,C;YAAwB,oB  
AAO,O;YAAP,sB;;;QAE5B,oBAAO,I;;;MA/VP,wB;K;IAGJ,6B;MAOQ,kBADE,SACF,Q;QAAW,OAAy,SAAL,  
SAAK,C;;QAEhB,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAa,C;UACI,MAAM,2BAaUaB,sBAaVb,C;QACV,OA  
AO,QAAS,O;;K;IAK5B,6B;MAMI,IAAI,mBAAJ,C;QACI,MAAM,2BAaUaB,gBAaVb,C;MACV,OAAO,sBAAK,  
CAAL,C;K;mFAGX,yB;MAAA,iE;MAAA,uC;QAKoB,Q;QAAA,2B;QAaHb,OAaGb,cAAhB,C;UAaGb,yB;UA  
AM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAaUaB,wDAaVb,C;O;KANV,C;oGAS  
A,yB;MAAA,iE;MAAA,uC;QASW,Q;QAAA,+B;;UAYS,U;UAAA,6B;UAaHb,OAaGb,gBAaHb,C;YAAgB,2B;  
YACZ,aAbwB,SAaX,CAAU,OAAV,C;YACb,IAAI,cAAJ,C;cACI,8BAAO,M;cAAP,gC;;;UAGR,8BAAO,I;;;QAI  
BA,kC;QAAA,iB;UAAmC,MAAM,gCAaUaB,mEAAvB,C;;QAaHd,OAAO,I;O;KATX,C;gHAYA,gC;MASoB,Q;  
MAAA,2B;MAaHb,OAaGb,cAAhB,C;QAaGb,yB;QACZ,aAAa,UAAU,OAAV,C;QACb,IAAI,cAAJ,C;UACI,O  
AAO,M;;MAGf,OAAO,I;K;IAGX,mC;MAKQ,kBADE,SACF,Q;QACI,IAAI,mBAAJ,C;UACI,OAAO,I;;UAEP,O  
AAO,sBAAK,CAAL,C;;QAGX,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAa,C;UACI,OAAO,I;QACX,OAAO,QA  
AS,O;;K;IAK5B,mC;MAII,OAAW,mBAAJ,GAAe,IAAf,GAAYb,sBAAK,CAAL,C;K;+FAGpC,gC;MAIoB,Q;MA  
AA,2B;MAaHb,OAaGb,cAAhB,C;QAaGb,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;MACr  
D,OAAO,I;K;0FAGX,yB;MAAA,8D;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAaC,sBA  
AI,KAJ,CAAtC,GAAaD,aAAa,KAAb,C;O;KALjE,C;IAQA,uC;MAMI,OAAW,SAAS,CAAT,IAAc,SAAS,2BA  
A3B,GAAaC,sBAAI,KAJ,CAAtC,GAAaD,I;K;IAGjE,uC;MAMIb,Q;MAFb,IAAI,8BAAJ,C;QAaKB,OAAO,SA  
AK,eAAQ,OAAR,C;MAC9B,YAAy,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,mBAaMb,KAAnB,C  
;QACA,IAAI,gBAAW,IAAX,CAAJ,C;UACI,OAAO,K;QACX,qB;;MAEJ,OAAO,E;K;IAGX,uC;MAKI,OAAO,w  
BAAQ,OAAR,C;K;gGAGX,yB;MAAA,wE;MAAA,uC;QAKiB,Q;QADb,YAAy,C;QACC,2B;QAAb,OAAa,cAA  
b,C;UAAa,sB;UACT,mBAaMb,KAAnB,C;UACA,IAAI,UAAU,IAAV,CAAJ,C;YACI,OAAO,K;UACX,qB;;QAE  
J,OAAO,E;O;KAXX,C;gGAcA,gC;MAKiB,Q;MADb,YAAy,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAC  
T,IAAI,UAAU,IAAV,CAAJ,C;UACI,OAAO,K;QACX,qB;;MAEJ,OAAO,E;K;8FAGX,yB;MAAA,wE;MAAA,uC  
;QAMiB,Q;QAFb,gBAaGb,E;QACHb,YAAy,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,mBAaMb,K  
AAnB,C;UACA,IAAI,UAAU,IAAV,CAAJ,C;YACI,YAAy,K;UACHb,qB;;QAEJ,OAAO,S;O;KAZX,C;8FAeA,g  
C;MAII,eAAe,SAAK,sBAaA,cAAb,C;MACpB,OAAO,QAAS,cAAhB,C;QACI,IAAI,UAAU,QAAS,WAAAnB,CA  
AJ,C;UACI,OAAO,QAAS,Y;;MAGxB,OAAO,E;K;IAGX,4B;MASQ,kBADE,SACF,Q;QAAW,OAAy,QAAL,S  
AAK,C;;QAEhB,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAa,C;UACI,MAAM,2BAaUaB,sBAaVb,C;QACV,WAA  
W,QAAS,O;QACpB,OAAO,QAAS,UAAhB,C;UACI,OAAO,QAAS,O;QACpB,OAAO,I;;K;IAKnB,4B;MAQI,IA  
AI,mBAAJ,C;QACI,MAAM,2BAaUaB,gBAaVb,C;MACV,OAAO,sBAAK,2BAAL,C;K;iFAGX,yB;MAAA,iE;M  
AAA,gB;MAAA,8B;MAAA,uC;QAUoB,UAQT,M;QAVP,WAAe,I;QACf,YAAy,K;QACI,2B;QAaHb,OAaGb,c  
AAhB,C;UAaGb,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;YACI,OAAO,O;YACP,QAAQ,I;;QAGhB,IAAI,CAA  
C,KAAL,C;UAAy,MAAM,gCAaUaB,wDAaVb,C;QAEIB,OAAO,2E;O;KAlBX,C;iFAqBA,yB;MAAA,iE;MAAA,  
uC;QAQI,eAAe,SAAK,sBAaA,cAAb,C;QACpB,OAAO,QAAS,cAAhB,C;UACI,cAAc,QAAS,W;UACvB,IAAI,U  
AAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEhC,MAAM,gCAaUaB,kDAaVb,C;O;KAbV,C;IAGBA,2C;MAOiB,

Q;MAHb,IAAI,8BAAJ,C;QAAkB,OAAO,SAAK,mBAAY,OAAZ,C;MAC9B,gBAAgB,E;MACHb,YAAY,C;MA  
CC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,mBAAmB,KAAhB,C;QACA,IAAI,gBAAW,IAAX,CAAJ,C;UAC  
I,YAAY,K;QACHb,qB;;MAEJ,OAAO,S;K;IAGX,2C;MAKI,OAAO,4BAAY,OAAZ,C;K;IAGX,kC;MAOQ,kBAD  
E,SACF,Q;QAAW,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,iBAAO,CAAP,IAAL,C;;QAEvC,eAAe,oB;QA  
Cf,IAAI,CAAC,QAAS,UAAAd,C;UACI,OAAO,I;QACX,WAAW,QAAS,O;QACpB,OAAO,QAAS,UAAhB,C;UA  
CI,OAAO,QAAS,O;QACpB,OAAO,I;;K;IAKnB,kC;MAMI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,iBAA  
O,CAAP,IAAL,C;K;6FAGpC,gC;MAOoB,Q;MADhB,WAAe,I;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,y  
B;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,OAAO,O;;MAGf,OAAO,I;K;6FAGX,gC;MAMI,eAAe,SAAK,sBA  
Aa,cAAb,C;MACpB,OAAO,QAAS,cAAhB,C;QACI,cAAc,QAAS,W;QACvB,IAAI,UAAU,OAAV,CAAJ,C;UAA  
wB,OAAO,O;;MAEnC,OAAO,I;K;qFAGX,yB;MAAA,mC;MAAA,gD;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C  
;O;KARX,C;IAWA,sC;MAOI,IAAI,mBAAJ,C;QACI,MAAM,2BAAUb,sBAAvB,C;MACV,OAAO,qBAAU,MAA  
O,iBAAQ,cAAR,CAAjB,C;K;iGAGX,yB;MAAA,mC;MAAA,4D;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;K  
APX,C;IAUA,4C;MAMI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAO,qBAAU,MAAO,iBAAQ,cAAR,CAAjB,  
C;K;IAGX,8B;MAKQ,kBADE,SACF,Q;QAAW,OAAy,UAAAL,SAAK,C;;QAEhB,eAAe,oB;QACf,IAAI,CAAC,Q  
AAS,UAAAd,C;UACI,MAAM,2BAAUb,sBAAvB,C;QACV,aAAa,QAAS,O;QACTb,IAAI,QAAS,UAAb,C;UACI,  
MAAM,gCAAyB,uCAAZB,C;QACV,OAAO,M;;K;IAKnB,8B;MALiB,IAAN,I;MAAA,QAAM,cAAN,C;aACH,C;  
UAAK,MAAM,2BAAUb,gBAAvB,C;aACX,C;UAAK,6BAAK,CAAL,C;UAAAL,K;;UACQ,MAAM,gCAAyB,iCA  
AzB,C;;MAHIB,W;K;qFAOJ,yB;MAAA,kF;MAAA,iE;MAAA,gB;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QA  
XP,aAAiB,I;QACjB,YAAY,K;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CA  
AJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAYb,qDAAZB,C;YACjB,SAAS,O;YACT,QAAQ,I;;QAGhB,IAAI  
,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,wDAAvB,C;QAEIB,OAAO,6E;O;KafX,C;IAkBA,oC;MAKQ,kBADE  
,SACF,Q;QAAW,OAAW,mBAAQ,CAAZ,GAAe,sBAAK,CAAL,CAAF,GAA4B,I;;QAE1C,eAAe,oB;QACf,IAAI,  
CAAC,QAAS,UAAAd,C;UACI,OAAO,I;QACX,aAAa,QAAS,O;QACTb,IAAI,QAAS,UAAb,C;UACI,OAAO,I;QA  
CX,OAAO,M;;K;IAKnB,oC;MAII,OAAW,mBAAQ,CAAZ,GAAe,sBAAK,CAAL,CAAF,GAA4B,I;K;iGAGvC,g  
C;MAMoB,Q;MAFhB,aAAiB,I;MACjB,YAAY,K;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI  
,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;MAGhB,IAAI,  
CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;IAGX,8B;MAoBsC,UAGT,MAHS,EAarB,M;MN/pBb,IA  
AI,EMsoBI,KAAK,CNtoBT,CAAJ,C;QACI,cMqoBc,sD;QNpoBd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MMqoBV  
,IAAI,MAAK,CAAT,C;QAAY,OAAO,mB;MACnB,Q;MACA,IAAI,oCAAJ,C;QACI,iBAAiB,iBAAO,CAAP,I;Q  
ACjB,IAAI,cAAc,CAAIB,C;UACI,OAAO,W;QACX,IAAI,eAAc,CAAIB,C;UACI,OAAO,OAAO,kBAAP,C;QAC  
X,OAAO,iBAAa,UAAb,C;QACP,IAAI,8BAAJ,C;UACI,IAAI,sCAAJ,C;YAC0B,qB;YAAtB,iBAAc,CAAd,wB;cA  
CI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;;YAEI,wCAAA,CAAb,C;YAAb,OAAa,gBAAb,C;cAAa,wB;cACT,IAA  
K,WAAI,IAAJ,C;;UAEb,OAAO,I;;QAIX,OAAO,gB;;MAEX,YAAY,C;MACC,6B;MAAb,OAAa,gBAAb,C;QA  
Aa,0B;QACT,IAAI,SAAS,CAAb,C;UAAgB,IAAK,WAAI,MAAJ,C;;UAAe,qB;;MAEXC,OAAy,qBAAL,IAAK,C  
;K;IAGhB,kC;MNrqBI,IAAI,EM6qBI,KAAK,CN7qBT,CAAJ,C;QACI,cM4qBc,sD;QN3qBd,MAAM,gCAAyB,O  
AAQ,WAAjC,C;;MM4qBV,OAAO,kBAAgB,gBAAV,iBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;kGAGX,y  
B;MAAA,4C;MAAA,qD;MAAA,uC;QAMI,IAAI,CAAC,mBAAL,C;UACI,eAAe,+BAAa,cAAb,C;UACf,OAAO,  
QAAS,cAAhB,C;YACI,IAAI,CAAC,UAAU,QAAS,WAAhB,CAAL,C;cACI,OAAO,gBAAK,QAAS,YAAT,GAA  
uB,CAAvB,IAAL,C;;;QAIhB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,uC;QAQiB,Q;QAFb,eAAe,  
K;QACf,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,QA AJ,C;YACI,IAAK,WAAI,IAAJ,  
C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;QAEhB,OAAO,I;O;  
KafX,C;oFAkBA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QA2FA,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,  
C;UAAgB,yB;UAAM,IA3FU,SA2FN,CAAU,OAAV,CAAJ,C;YAAwB,WAAy,WAAI,OAAJ,C;;QA3FID,OA4F  
O,W;O;KAIGX,C;kGASA,yB;MAAA,+D;MA6jCA,wE;MA7jCA,uC;QAQW,kBAAgB,gB;QA4jCV,gB;QADb,Y  
AAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAhjCT,IAZmC,SAY/B,CAGjCkB,oBAAmB,cAAnB,EAAM  
B,sBAAhB,UAhjCIB,EAjC+C,IAhjC/C,CAAJ,C;YAA2C,sBAjCQ,IAhjCR,C;;QAZ/C,OAco,W;O;KATBX,C;sG  
AWA,yB;MAKjCA,wE;MALjCA,oD;QAYjCiB,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;U  
AhjCT,IAAI,UAgjCkB,oBAAmB,cAAnB,EAAMB,sBAAhB,UAhjCIB,EAjC+C,IAhjC/C,CAAJ,C;YAA2C,sBAg

jCQ,IAhjCR,C;;QAE/C,OAAO,W;O;KAXX,C;wGAcA,yB;MAAA,+D;MAAA,sC;QAMW,kBAAmB,gB;QASV,Q  
;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,YAAJ,C;YAAkB,WAAy,WAAI,OAAJ,C;;QAT  
pD,OAuO,W;O;KAhBX,C;4GASA,4C;MAMoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,I  
AAI,YAAJ,C;UAAkB,WAAy,WAAI,OAAJ,C;;MACpD,OAAO,W;K;0FAGX,yB;MAAA,+D;MAAA,uC;QAMW  
,kBAAY,gB;QA4BH,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CA5BS,SA4BR,CAAU,  
OAAV,CAAL,C;YAAyB,WAAy,WAAI,OAAJ,C;;QA5B3D,OA6BO,W;O;KAnCX,C;IASA,oC;MAMI,OAAO,6  
BAAgB,gBAAhB,C;K;IAGX,mD;MAMoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,e  
AAJ,C;UAAqB,WAAy,WAAI,OAAJ,C;;MACvD,OAAO,W;K;8FAGX,6C;MAMoB,Q;MAAA,2B;MAAhB,OAA  
gB,cAAhB,C;QAAgB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAy,WAAI,OAAJ,C;;MAC3  
D,OAAO,W;K;wFAGX,6C;MAMoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,  
OAAV,CAAJ,C;UAAwB,WAAy,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;IAGX,sC;MAII,IAAI,OAAQ,UAAZ,C;  
QAAuB,OhB7wBe,W;;MgB8wBtC,OAA6D,SAAtD,SAAK,iBAAQ,OAAQ,MAAhB,EAAuB,OAAQ,aAAR,GAA  
uB,CAAvB,IAAvB,CAAI,D,C;K;IAGjE,sC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MA  
CnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACtB,WAAW,iBAAa,IAAb,C;MACG,yB;MAAd,OAAc,cAAAd,C;  
QAAc,uB;QACV,IAAK,WAAI,sBAAL,KAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,8B;MAGBiB,Q;MN91Bb,IAAI,  
EMs1BI,KAAK,CNt1BT,CAAJ,C;QACI,cMq1Bc,sD;QNp1Bd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MMq1BV,IA  
AI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,oCAAJ,C;QACI,IAAI,KAAK,cAAT,C;UAAe,OAAO,mB;Q  
ACtB,IAAI,MAAK,CAAT,C;UAAy,OAAO,OAAO,mBAAP,C;;MAEvB,YAAy,C;MACZ,WAAW,iBAAa,CAAb  
,C;MACE,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UAC  
I,K;;MAER,OAAy,qBAAL,IAAK,C;K;IAGhB,kC;MAeqC,IAGhB,I;MNx3BjB,IAAI,EM82BI,KAAK,CN92BT,C  
AAJ,C;QACI,cM62Bc,sD;QN52Bd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MM62BV,IAAI,MAAK,CAAT,C;QAA  
Y,OAAO,W;MACnB,WAAW,c;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;  
QAAY,OAAO,OAAO,kBAAP,C;MACnB,WAAW,iBAAa,CAAb,C;MACX,IAAI,sCAAJ,C;QACI,iBAAc,OAAO,  
CAAP,IAAd,UAA6B,IAA7B,U;UACI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;;QAEI,sCAAa,OAAO,CAAP,IAAb,  
C;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAK,WAAI,IAAJ,C;;MAEb,OAAO,I;K;kGAGX,yB;MAAA,qD;MA  
AA,gE;MAAA,gD;MAAA,uC;QAMI,IAAI,mBAAJ,C;UACI,OAAO,W;QACX,eAAe,+BAAa,cAAb,C;QACf,OA  
AO,QAAS,cAAhB,C;UACI,IAAI,CAAC,UAAU,QAAS,WAAAnB,CAAL,C;YACI,QAAS,O;YACT,mBAAmB,iBA  
AO,QAAS,YAAhB,I;YACnB,IAAI,iBAAgB,CAApB,C;cAAuB,OAAO,W;YACI,kBAA3B,eAAa,YAAb,C;YACH  
,OAAgB,kBAAhB,C;cACI,sBAAa,eAAb,C;YAFR,OH51BD,W;;;QGk2BP,OAAO,iB;O;KApBX,C;0FAuBA,yB;  
MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,CA  
AC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO,I;O;KAZX,C;IAoBA,+B;MAII,I  
AAI,wCAAsB,kBAAQ,CAAIC,C;QAAqC,OAAO,mB;MAC5C,WAAW,0B;MACN,WAAI,IAAK,C;MACL,OAA  
O,I;K;IAGX,uC;MAOI,aAAU,2BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;Q  
ACf,sBAAK,CAAL,EAAU,SAAK,aAAI,CAAJ,EAAO,sBAAK,CAAL,CAAP,CAAf,C;;K;oFAIR,yB;MAAA,oD;  
MJr4BA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,c  
AAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MI83Bf,sC;QAMI,IAAI,iBA  
AO,CAAX,C;UAAc,oBJp4Bd,eAAW,iBIo4BsB,QJp4BtB,CAAX,CIo4Bc,C;;O;KANIB,C;wGASA,yB;MAAA,oD;  
MJ33BA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,c  
AAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MIo3Bf,sC;QAMI,IAAI,iBAO  
,CAAX,C;UAAc,oBJ13Bd,eAAW,2BI03BgC,QJ13BhC,CAAX,CIo3Bc,C;;O;KANIB,C;IASA,sC;MAMI,sBAAS,c  
AAT,C;K;IAGJ,6B;MASgB,Q;MAHZ,IAAI,oCAAJ,C;QACI,IAAI,kBAAQ,CAAZ,C;UAAe,OAAy,SAAL,SAAK  
,C;QAEwB,kBAA3C,sBC7+Bsd,sBD6+BtD,uB;QAAMd,mB;QAA3D,OAAoE,OHp7BjE,WGo7BiE,C;;MAEjD,k  
BAAhB,0B;MAAwB,oB;MAA/B,OHt7BO,W;K;wFGy7BX,yB;MAAA,wD;MJ96BA,sC;MAAA,oC;MAAA,uBA  
Oe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA  
2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MIu6Bf,sC;QAQI,OAAO,sBJ/6BP,eAAW,iBI+6BiB,QJ/6BjB,CA  
AX,Ci+6BO,C;O;KARX,C;4GAWA,yB;MAAA,wD;MJt6BA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe  
,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3  
B,C;W;S;OA+EI,C;MI+5Bf,sC;QAMI,OAAO,sBJr6BP,eAAW,2BIq6B2B,QJr6B3B,CAAX,CiQ6BO,C;O;KANX,

C;IASA,uC;MAMI,OAAO,wBAAW,cAAX,C;K;IAGX,6C;MASE,Q;MAHX,IAAI,oCAAJ,C;QACG,IAAI,kBAAQ,CAAZ,C;UAAe,OAAy,SAAL,SAAK,C;QAEe,kBAaIC,sBCxhCuD,sBDwhCvD,uB;QAA0C,iC;QAAID,OAAyE,OH/9BrE,WG+9BqE,C;;MAErD,kBAAhB,0B;MAAwB,mC;MAA/B,OHj+BO,W;K;IGo+BX,qC;MAMoB,UACL,M;MAHX,aAAa,oBAAa,cAAb,C;MACb,YAAy,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OA AO,cAAP,EAAO,sBAAP,YAAkB,O;;MACTB,OAAO,M;K;IAGX,kC;MAMoB,UACL,M;MAHX,aAAa,cAAU,cAA V,C;MACb,YAAy,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,cAAP,EAAO,sBAAP,Y AAkB,O;;MACTB,OAAO,M;K;IAGX,kC;MAMoB,UACL,M;MAHX,aAAa,iBAAU,cAAV,C;MACb,YAAy,C;M ACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,oC;QACZ,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACTB,OAAO, M;K;IAGX,oC;MAMoB,UACL,M;MAHX,aAAa,iBAAy,cAAZ,C;MACb,YAAy,C;MACI,2B;MAAhB,OAAgB,c AAhB,C;QAAgB,yB;QACZ,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACTB,OAAO,M;K;IAGX,mC;MAMoB,U ACL,M;MAHX,aAAa,iBAAW,cAAX,C;MACb,YAAy,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QA CZ,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACTB,OAAO,M;K;IAGX,iC;MAMoB,UACL,M;MAHX,aAAa,eA AS,cAAT,C;MACb,YAAy,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,cAAP,EAAO,sB AAP,YAAkB,O;;MACTB,OAAO,M;K;IAGX,kC;MAMoB,UACL,M;MAHX,aAAa,iBAAU,cAAV,C;MACb,YAA Y,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACTB, OAAO,M;K;IAGX,mC;MAMoB,UACL,M;MAHX,aAAa,eAAW,cAAX,C;MACb,YAAy,C;MACI,2B;MAAhB,O AAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACTB,OAAO,M;K;0FAGX,yB;M AAA,kF;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAwD,cAAzC,YAAy,mCAAwB,EAAXB,CAAZ, CAAyC,EAAC,EAAD,C;QACjD,kBAAY,mBAAoB,QAAPB,C;QAYEH,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C; UAAgB,yB;UACZ,WA1E8C,SA0E/B,CAAU,OAAV,C;UfPkBnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAArB,C ;;Qe0fA,OA4EO,W;O;KAXFX,C;+FAeA,yB;MAAA,kF;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eA AwD,cAAzC,YAAy,mCAAwB,EAAXB,CAAZ,CAAyC,EAAC,EAAD,C;QACjD,kBAAC,mBAAoB,QAAPB,C;QA 2BL,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAy,aA5BoC,WA4BhC,CAAy,OAAZ,CAA J,EA0B,OAA1B,C;;QA5BhB,OA8BO,W;O;KA1CX,C;+FAeA,yB;MAAA,kF;MAAA,0D;MAAA,yD;MAAA,uE ;MAAA,yD;QAUl,eAAwD,cAAzC,YAAy,mCAAwB,EAAXB,CAAZ,CAAyC,EAAC,EAAD,C;QACjD,kBAAC,m BAAoB,QAAPB,C;QA6BL,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAy,aA9BoC,WA8B hC,CAAy,OAAZ,CAAJ,EA9BiD,cA8BvB,CAAE,OAaf,CAA1B,C;;QA9BhB,OA9CO,W;O;KA3CX,C;mGAcA,+ C;MAUoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,WAAy,aAAI,YAAy,OAAZ,CAAJ,EA A0B,OAA1B,C;;MAEHb,OAAO,W;K;mGAGX,+D;MAUoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,y B;QACZ,WAAy,aAAI,YAAy,OAAZ,CAAJ,EA0B,eAAe,OAaf,CAA1B,C;;MAEHb,OAAO,W;K;8FAGX,6C; MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,WAAe,UAAU,OAAV,C;QfPkBnB,wBAAI, IA AK,MAAT,EAAGB,IAAK,OAArB,C;;MeskBA,OAAO,W;K;kGAGX,yB;MAAA,kF;MAAA,0D;MAAA,yD;M AAA,uE;MAAA,2C;QAYl,aAAa,mBAA6D,cAAzC,YAAy,mCAAwB,EAAXB,CAAZ,CAAyC,EAAC,EAAD,CAA 7D,C;QACG,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAbO,MAcP,aAAI,OAAJ,EAde,aAcF,CAAc, OAAAd,CAAb,C;;QAdhB,OAAuB,M;O;KAb3B,C;sGAgBA,iD;MAUoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C; QAAgB,yB;QACZ,WAAy,aAAI,OAAJ,EAaA,cAAc,OAAAd,CAAb,C;;MAEHb,OAAO,W;K;IAGX,gD;MAIiB,Q; MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAy,WAAI,IAAJ,C;;MAEHb,OAAO,W;K;IAGX,gC;MAI l,OAAO,0BAAa,eAAW,YAAy,mCAAwB,EAAXB,CAAZ,CAAX,CAAb,C;K;IAGX,6B;MAKqB,IAAN,I;MADX, IAAI,oCAAJ,C;QACW,QAAM,cAAN,C;eACH,C;YAAK,kB;YAAL,K;eACA,C;YAAK,cAAW,8BAAJ,GAAkB,s BAAI,CAAJ,CAAIB,GAA8B,oBAAW,OAAhD,C;YAAL,K;;YACa,uBAAL,SAAK,C;YAHV,K;;QAAP,W;;MAM J,OAA4B,qBAAhB,gBAAL,SAAK,CAAgB,C;K;IAGhC,oC;MAII,IAAI,oCAAJ,C;QACI,OAAy,gBAAL,SAAK, C;MACHb,OAAO,0BAAa,gBAAb,C;K;IAGX,oC;MAII,OAAO,iBAAU,SAAV,C;K;IAGX,4B;MAOqB,IAAN,I;M ADX,IAAI,oCAAJ,C;QACW,QAAM,cAAN,C;eACH,C;YAAK,iB;YAAL,K;eACA,C;YAAK,aAAU,8BAAJ,GAA kB,sBAAK,CAAL,CAAIB,GAA+B,oBAAW,OAAhD,C;YAAL,K;;YACQ,iCAAa,qBAAiB,YAAy,cAAZ,CAAjB, CAAb,C;YAHL,K;;QAAP,W;;MAMJ,OAAwC,oBAAjC,0BAAa,sBAAb,CAAiC,C;K;sFAG5C,yB;MAAA,+D;M AwFA,gD;MAxFA,uC;QAMW,kBAAU,gB;QASFD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ, WAvF6B,SAuFIB,CAAU,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAXfHb,OA0FO,W;O;KAhGX,C;uF ASA,yB;MAAA,+D;MA0FA,gD;MA1FA,uC;QAUW,kBAAU,gB;QAWFD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB

,C;UAAgB,yB;UACZ,WaZf6B,SAyFIB,CAAU,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QA1FhB,OA4F  
O,W;O;KAtGX,C;oGAaA,yB;MAAA,+D;MA8BA,wE;MAAA,gD;MA9BA,uC;QAYW,kBAaIB,gB;QA6BR,gB;  
QADhB,YAAY,C;QACI,2B;QAaHb,OAAGb,cAAhB,C;UAAgB,yB;UACZ,WA9BoC,SA8BzB,CAAU,oBAAmB,  
cAAnB,EAAMb,sBAAnB,UAAV,EAAuC,OAAvC,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QA/BhB,OAiCO,W;  
O;KA7CX,C;oGAeA,yB;MAAA,+D;MAiCA,wE;MAAA,gD;MAjCA,uC;QAYW,kBAaIB,gB;QAgCR,gB;QADh  
B,YAAY,C;QACI,2B;QAaHb,OAAGb,cAAhB,C;UAAgB,yB;UACZ,WajCoC,SAiCzB,CAAU,oBAAmB,cAAnB  
,EAAMb,sBAAnB,UAAV,EAAuC,OAAvC,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAiChB,OAoCO,W;O;KA  
hDX,C;wGAeA,yB;MAAA,wE;MAAA,gD;MAAA,oD;QAWoB,UAC4B,M;QAF5C,YAAY,C;QACI,2B;QAaHb,  
OAAGb,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,oBAAmB,cAAnB,EAAMb,sBAAnB,UAAV,EAAuC,OAA  
vC,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;yGakBA,yB;MAAA,wE;MAAA,gD;  
MAAA,oD;QAWoB,UAC4B,M;QAF5C,YAAY,C;QACI,2B;QAaHb,OAAGb,cAAhB,C;UAAgB,yB;UACZ,WAA  
W,UAAU,oBAAmB,cAAnB,EAAMb,sBAAnB,UAAV,EAAuC,OAAvC,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;  
;QAEhB,OAAO,W;O;KafX,C;0FAkBA,yB;MAAA,gD;MAAA,oD;QAiOB,Q;QAAA,2B;QAaHb,OAAGb,cAAhB  
,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KA  
RX,C;2FAWA,yB;MAAA,gD;MAAA,oD;QAQoB,Q;QAAA,2B;QAaHb,OAAGb,cAAhB,C;UAAgB,yB;UACZ,  
WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAZX,C;uFAeA,yB;MA  
AA,wE;MAyBA,+D;MAzBA,yC;QASW,kBAAU,oB;QAYBD,Q;QAAA,2B;QAaHb,OAAGb,cAAhB,C;UAAgB,y  
B;UACZ,UA1BiD,WA0BvC,CAAY,OAAZ,C;UfVnCP,U;UADP,YeynCe,WfznCH,WeynCwB,GfznCxB,C;UACL,  
IAAI,aAAJ,C;YACH,aeunCuC,gB;YAA5B,WftnCX,aesnCgC,GftnChC,EAAS,MAAT,C;YACA,e;;YAEA,c;;Uem  
nCA,iB;UACA,IAAK,WAAI,OAAJ,C;;QA5BT,OA8BO,W;O;KAvcX,C;uFAYA,yB;MAAA,wE;MA8BA,+D;MA  
9BA,yD;QAUW,kBAAU,oB;QA8BD,Q;QAAA,2B;QAaHb,OAAGb,cAAhB,C;UAAgB,yB;UACZ,UA/BiD,WA+  
BvC,CAAY,OAAZ,C;UfzoCP,U;UADP,Ye2oCe,Wf3oCH,We2oCwB,Gf3oCxB,C;UACL,IAAI,aAAJ,C;YACH,ae  
yoCuC,gB;YAA5B,WfxoCX,aewoCgC,GfxoChC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UeqoCA,iB;UACA,IAAK,  
WajCyD,cAiCrD,CAAe,OAaf,CAAJ,C;;QAJCT,OAmCO,W;O;KA7CX,C;0FAaA,yB;MAAA,+D;MAAA,sD;QA  
SoB,Q;QAAA,2B;QAaHb,OAAGb,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;UfVnCP,U;UADP,Yey  
nCe,WfznCH,WeynCwB,GfznCxB,C;UACL,IAAI,aAAJ,C;YACH,aeunCuC,gB;YAA5B,WftnCX,aesnCgC,Gftn  
hC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UemnCA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KADx,C  
;2FAiBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAAA,2B;QAaHb,OAAGb,cAAhB,C;UAAgB,yB;UACZ,UAAU,  
YAAY,OAAZ,C;UfzoCP,U;UADP,Ye2oCe,Wf3oCH,We2oCwB,Gf3oCxB,C;UACL,IAAI,aAAJ,C;YACH,ae  
yoCuC,gB;YAA5B,WfxoCX,aewoCgC,GfxoChC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UeqoCA,iB;UACA,IAAK,  
WAAI,eAAe,OAaf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;4FAkBA,yB;MAAA,kC;MAAA,4C;MAAA,wE;QAQW,sC;  
QAAA,8C;O;MARX,oDASQ,Y;QAA6C,OAAA,oBAAGB,W;O;MATrE,iDAUQ,mB;QAAoC,gCAAY,OAAZ,C;O  
;MAV5C,gF;MAAA,yC;QAQI,2D;O;KARJ,C;8EAcA,yB;MAAA,kF;MAAA,gE;MAAA,uC;QAOW,kBAAM,eA  
Aa,mCAAwB,EAAXB,CAAb,C;QAuEA,Q;QAAA,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAy,WAXEwC,  
SAwEpC,CAAU,IAAV,CAAJ,C;;QAxehB,OAYEO,W;O;KAhFX,C;4FAUA,yB;MAAA,kF;MAAA,gE;MA+BA,w  
E;MA/BA,uC;QAOW,kBAaA,eAAa,mCAAwB,EAAXB,CAAb,C;QAgCP,gB;QADb,YAAY,C;QACC,2B;QAAb,  
OAAa,cAAb,C;UAAa,sB;UACT,WAAy,WajC+C,SAiC3C,CAAU,oBAAmB,cAAnB,EAAMb,sBAAnB,UAAV,  
EAAuC,IAAvC,CAAJ,C;;QajChB,OakCO,W;O;KazCX,C;0GAUA,yB;MAAA,+D;MAoSA,wE;MApSA,uC;QA  
OW,kBAAoB,gB;QAoSd,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UA1RsB,U;UAAA,cA  
VQ,SAUR,CA0RT,oBAAmB,cAAnB,EAAMb,sBAAnB,UA1RS,EA0RoB,IA1RpB,W;YAA6C,6B;;;QAVhF,OA  
WO,W;O;KAIBX,C;8GAUA,yB;MA0RA,wE;MA1RA,oD;QAiSiB,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,c  
AAb,C;UAAa,sB;UA1RsB,U;UAAA,wBA0RT,oBAAmB,cAAnB,EAAMb,sBAAnB,UA1RS,EA0RoB,IA1RpB,W  
;YAA6C,6B;;;QACHF,OAAO,W;O;KARX,C;+FAWA,yB;MAAA,wE;MAAA,oD;QAQiB,UACoC,M;QAFjD,YA  
AY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAy,WAAI,UAAU,oBAAmB,cAAnB,EAAMb,sBA  
AnB,UAAV,EAAuC,IAAvC,CAAJ,C;;QACHB,OAAO,W;O;KAVX,C;4FAaA,yB;MAAA,+D;MAAA,uC;QAOW,  
kBAaA,gB;QAWPJ,Q;QAAA,2B;QAaHb,OAAGb,cAAhB,C;UAAgB,yB;UAhPK,U;UAAA,cARe,SAQf,CagPQ,  
OAhPR,W;YAA6C,6B;;;QAR3D,OASO,W;O;KAhBX,C;gGAUA,yB;MAAA,oD;QAqPoB,Q;QAAA,2B;QAaHb,  
OAAGb,cAAhB,C;UAAgB,yB;UAhPK,U;UAAA,wBAGPQ,OAhPR,W;YAA6C,6B;;;QAC3D,OAAO,W;O;KANX

,C;kFASA,6C;MAKiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAy,WAAI,UAAU,IAAV,CAAJ ,C;;MACHB,OAAO,W;K;IAQiB,4C;MAAA,mB;QAAE,gC;O;K;IAL9B,gC;MAKI,OAAO,qBAaIB,6BAaJB,C;K; IAGX,+B;MASI,OAA2B,SAAf,eAAL,SAAK,CAAE,C;K;4FAG/B,yB;MAAA,2D;MAAA,+D;MAAA,sC;QAYc,Q ;QAFV,UAAU,c;QACV,WAAW,gB;QACD,2B;QAAV,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,SAAS,CAAT, C;UACV,IAAI,GAAI,WAAI,GAAJ,CAAR,C;YACI,IAAK,WAAI,CAAJ,C;;QAEb,OAAO,I;O;KAjBX,C;IAoBA, uC;MAQI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAI,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,sC;MAMI,UA Ae,eAAL,SAAK,C;MACX,YAAJ,GAAI,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,mC;MAMiB,IAAN,I;MACH ,kBADs,SACT,c;QAAoB,4BAAc,SAAd,C;;QACZ,iCAAA,sBAAb,C;MAFZ,W;K;IAMJ,mC;MAUI,UAAe,eAAL, SAAK,C;MACX,OAAJ,GAAI,EAAO,KAAp,C;MACJ,OAAO,G;K;8EAGX,yB;MAAA,gD;MAAA,uC;QAOoB,Q ;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,OAAO,I;QAC5B,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAA M,IAAI,CAAC,UAAU,OAAV,CAAL,C;YAAyB,OAAO,K;;QACtD,OAAO,I;O;KARX,C;IAWA,2B;MAMI,IAAI, oCAAJ,C;QAAwB,OAAO,CAAC,mB;MACHc,OAAO,oBAAW,U;K;+EAGtB,yB;MAAA,gD;MAAA,uC;QAOoB ,Q;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,OAAO,K;QAC5B,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;U AAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,I;;QACrD,OAAO,K;O;KARX,C;IAWA,6B;MAMoB,Q;MA FhB,IAAI,oCAAJ,C;QAAwB,OAAO,c;MAC/B,YAAy,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QA AM,oBAAmB,qBAAnB,EAAmB,KAAmB,E;;MACtB,OAAO,K;K;mFAGX,qB;MAKI,OAAO,c;K;mFAGX,yB;M AAA,gD;MAAA,wE;MAAA,uC;QAMoB,Q;QAFhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,OAAO,C;QAC5C,YA AY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,oBAAmB ,qBAAnB,EAAmB,KAAmB,E;;QAC9C,OAAO,K;O;KAPX,C;gFAUA,yC;MAUoB,Q;MADhB,kBAAkB,O;MACF ,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W ;K;8FAGX,yB;MAAA,wE;MAAA,gD;QAYoB,UAAiD,M;QAFjE,YAAy,C;QACZ,kBAAkB,O;QACF,2B;QAAh B,OAAgB,cAAhB,C;UAAgB,yB;UAAM,cAAc,UAAU,oBAAmB,cAAmB,EAAmB,sBAAnB,UAAV,EAAuC,WA AvC,EAAoD,OAApD,C;;QACpC,OAAO,W;O;KAbX,C;0FAgBA,yC;MASI,kBAAkB,O;MACIB,IAAI,CAAC,mB AAL,C;QACI,eAAe,+BAaA,cAAb,C;QACf,OAAO,QAAS,cAAhB,C;UACI,cAAc,UAAU,QAAS,WAAmB,EAA+ B,WAA/B,C;;MAGtB,OAAO,W;K;wGAGX,yC;MAUI,kBAAkB,O;MACIB,IAAI,CAAC,mBAAL,C;QACI,eAAe ,+BAaA,cAAb,C;QACf,OAAO,QAAS,cAAhB,C;UACI,YAAy,QAAS,gB;UACrB,cAAc,UAAU,KAAV,EAAiB,Q AAS,WAA1B,EAAc,WAAtC,C;;MAGtB,OAAO,W;K;sFAGX,6B;MAKoB,Q;MAAA,2B;MAAhB,OAAgB,cA AhB,C;QAAgB,yB;QAAM,OAAO,OAAP,C;;K;oGAG1B,yB;MAAA,wE;MAAA,oC;QAOiB,UAAgC,M;QAD7C, YAAy,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAAM,OAAO,oBAAmB,cAAmB,EAAmB,sBAAnB,UAAP ,EAAoC,IAAP,C,C;;O;KAPvB,C;IAUA,0B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,MAAM ,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MFxdDG,MAA O,KEwwDE,GFxdDF,EEwwDO,CFxdDP,C;;ME0wDd,OAAO,G;K;IAGX,2B;MAWI,eAAe,oB;MACf,IAAI,CA AC,QAAS,UAAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAA Q,QAAS,O;QACjB,MFxyDG,MAAO,KEwyDE,GFxyDF,EEwyDO,CFxyDP,C;;ME0yDd,OAAO,G;K;IAGX,2B; MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OA AO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAEvB, OAAO,G;K;kFAGX,yB;MAAA,sE;MAAA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MA AM,6B;QAC/B,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,O;QACHc,eAAe,SAAS,O AAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,U AAU,C;YACV,WAAW,C;;QAED,QAAT,QAAS,W;QACIB,OAAO,O;O;KAXBX,C;8FA2BA,+B;MAOI,eAAe,o B;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,cAAc,QAAS,O;MACvB,IAAI,CAAC,QAAS,U AAd,C;QAAyB,OAAO,O;MACHc,eAAe,SAAS,OAAT,C;;QAEX,QAAQ,QAAS,O;QACjB,QAAQ,SAAS,CAAT, C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,UAAU,C;UACV,WAAW,C;;MAED,QAAT,QAAS,W;MACIB,O AAO,O;K;mFAGX,yB;MAAA,sE;MF/2DA,iB;ME+2DA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C; UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS, QAAS,OAAIB,C;UACR,WFz3DG,MAAO,KEy3DO,QFz3DP,EEy3DiB,CFz3DjB,C;;QE23Dd,OAAO,Q;O;KApB X,C;mFAuBa,yB;MAAA,sE;MFj5DA,iB;MEi5DA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAy B,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,

OAAIB,C;UACR,WF35DG,MAAO,KE25DO,QF35DP,EE25DiB,CF35DjB,C;;QE65Dd,OAAO,Q;O;KApBX,C;m  
FAuBA,yB;MAAA,sE;MAAA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC  
/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,I  
AAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;+FAuBA,yB;MFp7DA,iB;MEo  
7DA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QACHC,eAAe,SAAS,QAAS,OA  
AIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WF57DG,MAAO,KE47DO,  
QF57DP,EE47DiB,CF57DjB,C;;QE87Dd,OAAO,Q;O;KAIbX,C;+FAqBA,yB;MFp9DA,iB;MEo9DA,sC;QAWI,e  
AAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QACHC,eAAe,SAAS,QAAS,OAAIB,C;QACf,OA  
AO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WF59DG,MAAO,KE49DO,QF59DP,EE49Di  
B,CF59DjB,C;;QE89Dd,OAAO,Q;O;KAIbX,C;+FAqBA,+B;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C  
;QAAyB,OAAO,I;MACHC,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,  
QAAS,OAAIB,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;0FAGX,yB;MA  
AA,sE;MAAA,kD;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS  
,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI,UAAW,S  
AAQ,QAAR,EAakB,CAAIB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;sGAu  
BA,2C;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,eAAe,SAAS,QAAS,OA  
AIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,UAAW,SAAQ,QAAR,E  
AAkB,CAAIB,CAAX,GAAkC,CAAtC,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;IAGX,gC;MAOI,eAAe,oB;MA  
Cf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;Q  
ACI,QAAQ,QAAS,O;QACjB,MFniEG,MAAO,KEmiEE,GFniEF,EEmiEO,CFniEP,C;;MEqiEd,OAAO,G;K;IAGX  
,iC;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,QAAS,O;MACnB,OA  
AO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MF/jEG,MAAO,KE+jEE,GF/jEF,EE+jEO,CF/jEP,C;;MEik  
Ed,OAAO,G;K;IAGX,iC;MAKI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,  
QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAAJ,C;UAA  
a,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0C;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,MAA  
M,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAAW,  
SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAKI  
,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,QAAS,O;MACnB,OAAO,QAA  
S,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;  
UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAy  
B,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MFt6  
DG,MAAO,KEs6DE,GFt6DF,EEs6DO,CFt6DP,C;;MEw6Dd,OAAO,G;K;IAGX,2B;MAWI,eAAe,oB;MACf,IAAI  
,CAAC,QAAS,UAAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,  
QAAQ,QAAS,O;QACjB,MFt8DG,MAAO,KEs8DE,GFt8DF,EEs8DO,CFt8DP,C;;MEw8Dd,OAAO,G;K;IAGX,2  
B;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,O  
AAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAEv  
B,OAAO,G;K;kFAGX,yB;MAAA,sE;MAAA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,M  
AAM,6B;QAC/B,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,O;QACHC,eAAe,SAAS,  
OAAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,  
UAAU,C;YACV,WAAW,C;;;QAED,QAAT,QAAS,W;QACIB,OAAO,O;O;KAXBX,C;8FA2BA,+B;MAOI,eAAe,  
oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,cAAc,QAAS,O;MACvB,IAAI,CAAC,QAAS,  
UAAAd,C;QAAyB,OAAO,O;MACHC,eAAe,SAAS,OAAAT,C;;QAEX,QAAQ,QAAS,O;QACjB,QAAQ,SAAS,CAA  
T,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,UAAU,C;UACV,WAAW,C;;;MAED,QAAT,QAAS,W;MACIB,  
OAAO,O;K;mFAGX,yB;MAAA,sE;MF7gEA,iB;ME6gEA,sC;QAAI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C  
;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,  
QAAS,OAAIB,C;UACR,WFvhEG,MAAO,KEuhEO,QFvhEP,EEuhEiB,CFvhEjB,C;;QEyhEd,OAAO,Q;O;KApBX  
,C;mFAuBA,yB;MAAA,sE;MF/iEA,iB;ME+iEA,sC;QAAI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,  
MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,O



AAIB,C;UACR,WFzjEG,MAAO,KEyjEO,QFzjEP,EEyjEiB,CFzjEjB,C;;QE2jEd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAAA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;+FAuBA,yB;MFIEA,iB;MEkIEA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WF1IEG,MAAO,KE0IEO,QF1IEP,EE0IEiB,CF1IEjB,C;;QE4IEd,OAAO,Q;O;KAIbX,C;+FAqBA,yB;MFInEA,iB;MEknEA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WF1nEG,MAAO,KE0nEO,QF1nEP,EE0nEiB,CF1nEjB,C;;QE4nEd,OAAO,Q;O;KAIbX,C;+FAqBA,+B;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;0FAGX,yB;MAAA,sE;MAAA,kD;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;sGAuBA,2C;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAAkC,CAAtC,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;IAGX,gC;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MFjsEG,MAAO,KEisEE,GFjsEF,EEisEO,CFjsEP,C;;MEmsEd,OAAO,G;K;IAGX,iC;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MF7tEG,MAAO,KE6tEE,GF7tEF,EE6tEO,CF7tEP,C;;ME+tEd,OAAO,G;K;IAGX,iC;MAKI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0C;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAAW,SAAQ,GAAR,EAaA,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAKI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAAW,SAAQ,GAAR,EAaA,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,4B;MAMI,IAAI,oCAAJ,C;QAAwB,OAAO,mB;MAC/B,OAAO,CAAC,oBAAW,U;K;iFAGvB,yB;MAAA,gD;MAAA,uC;QAOoB,Q;QADhB,IAAI,wCAAsB,mBAAIB,C;UAAqC,OAAO,I;QAC5B,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,K;;QACrD,OAAO,I;O;KARX,C;oFAWA,6B;MAKmC,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,OAAO,OAAP,C;;MAArC,gB;K;kGAGJ,yB;MAAA,6B;MAAA,sC;MA3wBA,wE;MA2wBA,2BAQIB,yB;QAnxBjB,wE;eAmxBiB,0B;UAAA,4B;YAAE,aAAe,c;YA5wBjB,gB;YADb,YAAY,C;YACC,2B;YAAb,OAAa,cAAb,C;cAAa,sB;cAAM,OAAO,oBAAmB,cAAnB,EAAMb,sBAAnB,UAAP,EAoC,IAApC,C;;YA4wBmB,W;W;S;OAAzB,C;MARjB,oC;QApwBiB,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAAM,OAAO,oBAAmB,cAAnB,EAAMb,sBAAnB,UAAP,EAoC,IAApC,C;;QA4wBnB,gB;O;KARJ,C;oFAWA,yB;MAAA,4F;MAAA,uC;QAaI,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,mCAA8B,oCAA9B,C;QAC/B,kBAAqB,QAAS,O;QAC9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,WAAV,EAAB,QAAS,OAAhC,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,wE;MAAA,uC;QAKBmD,Q;QAL/C,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,mCAA8B,oCAA9B,C;QAC/B,YAAY,C;QACZ,kBAAqB,QAAS,O;QAC9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,oBAAmB,YAAnB,EAAMb,oBAAnB,QAAM,EAAB,WAAvC,EAAD,QAAS,OAA7D,C;;QAEIB,OAAO,W;O;KApBX,C;8GAuBA,yB;MAAA,wE;MAAA,uC;QAKBmD,Q;QAL/C,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,YAAY,C;QACZ,kBAaQb,QAAS,O;QAC9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,oBAAmB,YAAnB,EAAMb,oBAAnB,QAAM,EAAB,WAAvC,EAAD,QAAS,OAA7D,C;;QAEIB,OAAO,W;O;KApBX,C;gGAuBA,gC;MAcI,eAAe,SAAK,W;MACpB,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,kBAAqB,QAAS,O;MAC9B,OAAO,QAAS,U

AAhB,C;QACI,cAAc,UAAU,WAAV,EAAuB,QAAS,OAAhC,C;;MAEIB,OAAO,W;K;8FAGX,yB;MAAA,4F;MAAA,uC;QAaI,eAAe,+BAAa,cAAb,C;QACf,IAAI,CAAC,QAAS,cAAAd,C;UACI,MAAM,mCAA8B,8BAA9B,C;QACV,kBAAqB,QAAS,W;QAC9B,OAAO,QAAS,cAAhB,C;UACI,cAAc,UAAU,QAAS,WAAAnB,EAA+B,WAA/B,C;;QAEIB,OAAO,W;O;KApBX,C;4GAuBA,yB;MAAA,4F;MAAA,uC;QAaI,eAAe,+BAAa,cAAb,C;QACf,IAAI,CAAC,QAAS,cAAAd,C;UACI,MAAM,mCAA8B,8BAA9B,C;QACV,kBAAqB,QAAS,W;QAC9B,OAAO,QAAS,cAAhB,C;UACI,YAAY,QAAS,gB;UACrB,cAAc,UAAU,KAaV,EAAiB,QAAS,WAA1B,EAAc,WAAtC,C;;QAEIB,OAAO,W;O;KArBX,C;wHAwBA,gC;MAaI,eAAe,+BAAa,cAAb,C;MACf,IAAI,CAAC,QAAS,cAAAd,C;QACI,OAAO,I;MACX,kBAAqB,QAAS,W;MAC9B,OAAO,QAAS,cAAhB,C;QACI,YAAY,QAAS,gB;QACrB,cAAc,UAAU,KAaV,EAAiB,QAAS,WAA1B,EAAc,WAAtC,C;;MAEIB,OAAO,W;K;0GAGX,gC;MAcI,eAAe,+BAAa,cAAb,C;MACf,IAAI,CAAC,QAAS,cAAAd,C;QACI,OAAO,I;MACX,kBAAqB,QAAS,W;MAC9B,OAAO,QAAS,cAAhB,C;QACI,cAAc,UAAU,QAAS,WAAAnB,EAA+B,WAA/B,C;;MAEIB,OAAO,W;K;8FAGX,yB;MAAA,kF;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAJhB,oBAAoB,mCAAwB,CAAxB,C;QACpB,IAAI,kBAAiB,CAArB,C;UAAwB,OAAO,OAAO,OAAP,C;QACc,kBAAhC,eAAa,gBAAGB,CAAhB,IAAb,C;QAAwC,8B;QAArD,aHzsFO,W;QG0sFP,kBAaKB,O;QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;4GAwBA,yB;MAAA,kF;MAAA,gD;MAAA,gE;MAAA,gD;QAmBoB,UACY,M;QAN5B,oBAAoB,mCAAwB,CAAxB,C;QACpB,IAAI,kBAAiB,CAArB,C;UAAwB,OAAO,OAAO,OAAP,C;QACc,kBAAhC,eAAa,gBAAGB,CAAhB,IAAb,C;QAAwC,8B;QAArD,aHluFO,W;QGmuFP,YAAY,C;QACZ,kBAaKB,O;QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,cAAV,EAAU,sBAaV,WAAmB,WAAAnB,EAAgC,OAAhC,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;kGA0BA,yB;MAAA,qD;MAAA,kF;MAAA,gE;MAAA,uC;QAcI,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,W;QACHC,sBAAqB,QAAS,OAA9B,C;QACuD,kBAA1C,eAAa,mCAAwB,EAAxB,CAAb,C;QAAkD,sBAAI,aAAJ,C;QAA/D,aH7vFO,W;QG8vFP,OAAO,QAAS,UAAhB,C;UACI,gBAAc,UAAU,aAAV,EAAuB,QAAS,OAAhC,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KATBX,C;gHAyBA,yB;MAAA,qD;MAAA,kF;MAAA,gE;MAAA,uC;QAOBgC,Q;QAN5B,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,W;QACHC,sBAAqB,QAAS,OAA9B,C;QACuD,kBAA1C,eAAa,mCAAwB,EAAxB,CAAb,C;QAAkD,sBAAI,aAAJ,C;QAA/D,aHtxFO,W;QGuxFP,YAAY,C;QACZ,OAAO,QAAS,UAAhB,C;UACI,gBAAC,WAAU,YAAV,EAAU,oBAAV,SAAmB,aAAnB,EAAgC,QAAS,OAAzC,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;gFA0BA,yB;MArGA,kF;MAAA,gD;MAAA,gE;MAqGA,gD;QAcW,sB;;UAI GS,Q;UAJhB,oBAAoB,mCAAwB,CAAxB,C;UACpB,IAAI,kBAAiB,CAArB,C;YAAwB,qBAAO,OAqGZ,OAryY,C;YAAP,uB;;UACqB,kBAAhC,eAAa,gBAAGB,CAAhB,IAAb,C;UAAwC,sBAoGIC,OApgkC,C;UAArD,aHzsFO,W;UG0sFP,kBAmgmB,O;UAIgh,2B;UAAhB,OAAgB,cAAhB,C;YAAgB,yB;YACZ,cAiGwB,SAjGV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;QA8FP,yB;O;KADJ,C;8FAiBA,yB;MA9FA,kF;MAAA,gD;MAAA,gE;MA8FA,gD;QAEW,6B;;UA1FS,gB;UALhB,oBAAoB,mCAAwB,CAAxB,C;UACpB,IAAI,kBAAiB,CAArB,C;YAAwB,4BAAO,OA8FL,OA9FK,C;YAAP,8B;;UACqB,kBAAhC,eAAa,gBAAGB,CAAhB,IAAb,C;UAAwC,sBA6F3B,OA7F2B,C;UAArD,aHluFO,W;UGmuFP,YAAY,C;UACZ,kBA2F0B,O;UA1FV,2B;UAAhB,OAAgB,cAAhB,C;YAAgB,yB;YACZ,cAyF+B,SAzFjB,EAAU,cAAV,EAAU,sBAaV,WAAmB,WAAAnB,EAAgC,OAAhC,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;QAsFP,gC;O;KAfJ,C;kFAkBA,+B;MAOoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;mFAGX,+B;MAUoB,Q;MADhB,UAAoB,C;MACJ,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;mFAGX,+B;MAUoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,yB;MAAA,SASoB,gB;MATpB,sC;QAUoB,Q;QADhB,Y;QACgB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;mFAgBA,yB;MIBvIFa,6B;MkBuIFa,sC;QAWoB,Q;QADhB,UIBvIFmC,ckBuIFnB,CIBvIFmB,C;QkBwIFnB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MIB35FiD,ckB25FjD,GIB35F2D,KAAK,GkB25FzD,SAAS,OAAT,CIB35FoE,KAAAX,IAAf,C;;QkB65FrD,OAAO,G;O;KADx,C;mFAiBA,yB;MDrmFA,+B;MCqmFA,sC;QAWoB,Q;QADhB,UDpmFqC,eAAW,oBComF/B,CDpmF+B,CAAX,C;QCqmFrB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAC

Z,MDz6FmD,eCy6FnD,GDz6F8D,KAAK,KCy6F5D,SAAS,OAAT,CDz6FuE,KAAx,CAAhB,C;;QC26FvD,OAA  
O,G;O;KAdX,C;IAiBA,qC;MAIoB,UAMT,M;MANS,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,eA  
AJ,C;UACI,MAAM,gCAAyB,2BAAwB,SAAxB,MAAzB,C;;MAId,OAAO,mE;K;IAGX,qC;MAIoB,UAMT,M;M  
ANS,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,eAAJ,C;UACI,MAAM,gCAAyB,2BAAwB,SAAxB  
,MAAzB,C;;MAId,OAAO,+D;K;IAGX,kC;MAWI,OAAO,oBAAS,IAAT,EAAe,IAAf,EAAc,IAAtC,C;K;IAGX,  
+C;MAgBI,OAAO,sBAAS,IAAT,EAAe,IAAf,EAAc,IAAtC,EAAwD,SAAXD,C;K;IAGX,mC;MAII,aAAa,iBAA  
a,mCAAwB,EAAxB,CAAb,C;MACb,kBAAc,KAAd,C;MAZuEgB,Q;MAAA,OA0uET,SA1uES,W;MAAhB,OA  
gB,cAAhB,C;QAAGB,2B;QAAU,oB;QA0uEK,IAAI,CAAC,SAAD,IAAY,OA1uEX,SA0uEW,UAAhB,C;UAAiC,  
YAAU,I;UAA3C,mBAAiD,K;;UAAjD,mBAA8D,I;;QA1uEvE,qB;UA0uED,MA1uEqC,WAAI,SAAJ,C;;MA0uE1  
D,OAAqB,M;K;IAGzB,sC;MAQI,IAAI,QrBypJG,YAAQ,CqBzpJf,C;QAAwB,OAAy,SAAL,SAAK,C;MACpC,Y  
AAqB,8BAAT,QAAS,C;MA5xEd,kBAAy,gB;MA4BH,Q;MAAA,OAiwET,SAjwES,W;MAAhB,OAAGB,cAAhB,  
C;QAAGB,yB;QAAM,IAAI,CAiwEF,qBAjwEa,OAiwEb,CAjwEF,C;UAAyB,WAAy,WAAI,OAAJ,C;;MAiwE3D  
,OAhwEO,W;K;IAMwEX,sC;MAQI,YAAqB,gCAAT,QAAS,EAAgC,SAAhC,C;MACrB,IAAI,KAAM,UAAV,C;  
QACI,OAAy,SAAL,SAAK,C;MA1yET,kBAAy,gB;MA4BH,Q;MAAA,OA+wET,SA/wES,W;MAAhB,OAAGB,c  
AAhB,C;QAAGB,yB;QAAM,IAAI,CA+wEF,qBA/wEa,OA+wEb,CA/wEF,C;UAAyB,WAAy,WAAI,OAAJ,C;;M  
A+wE3D,OA9wEO,W;K;IAixEX,sC;MAQI,YAAqB,8BAAT,QAAS,C;MACrB,IAAI,KAAM,UAAV,C;QACI,OA  
AY,SAAL,SAAK,C;MAXzET,kBAAy,gB;MA4BH,Q;MAAA,OA6xET,SA7xES,W;MAAhB,OAAGB,cAAhB,C;Q  
AAgB,yB;QAAM,IAAI,CA6xEF,qBA7xEa,OA6xEb,CA7xEF,C;UAAyB,WAAy,WAAI,OAAJ,C;;MA6xE3D,OA  
5xEO,W;K;8FA+xEX,yB;MAAA,8C;MAAA,qC;QAKI,OAAO,iBAAM,OAAN,C;O;KALX,C;0FAQA,yB;MAAA  
,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAy,gB;QACZ,aAAa,gB;QACG,2B;QAAhB,OAAGB,cAAhB,C  
;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;  
;;QAGf,OAAO,cAAK,KAAL,EAAy,MAAZ,C;O;KAjBX,C;IAoBA,kC;MAII,IAAI,oCAAJ,C;QAAwB,OAAy,O  
AAL,SAAK,EAAK,OAAL,C;MACpC,aAAa,gB;MACN,OAAP,MAAO,EAAO,SAAP,C;MACP,MAAO,WAAI,O  
AAJ,C;MACP,OAAO,M;K;IAGX,oC;MAII,aAAa,iBAAa,iBAAO,CAAP,IAAb,C;MACb,MAAO,gBAAO,SAAP,  
C;MACP,MAAO,WAAI,OAAJ,C;MACP,OAAO,M;K;IAGX,qC;MAII,IAAI,oCAAJ,C;QAAwB,OAAy,OAAL,S  
AAK,EAAK,QAAL,C;MACpC,aAAa,gB;MACN,OAAP,MAAO,EAAO,SAAP,C;MACA,SAAP,MAAO,EAAO,Q  
AAP,C;MACP,OAAO,M;K;IAGX,qC;MAII,aAAa,iBAAa,SAAK,KAAL,GAAY,QAAS,OAARB,IAAb,C;MACb,  
MAAO,gBAAO,SAAP,C;MACA,SAAP,MAAO,EAAO,QAAP,C;MACP,OAAO,M;K;IAGX,qC;MAII,IAAI,oCA  
AJ,C;QAAwB,OAAy,OAAL,SAAK,EAAK,QAAL,C;MACpC,aAAa,gB;MACN,OAAP,MAAO,EAAO,SAAP,C;  
MACA,OAAP,MAAO,EAAO,QAAP,C;MACP,OAAO,M;K;IAGX,qC;MAII,IAAI,mCAAJ,C;QACI,aAAa,iBAAa,  
SAAK,KAAL,GAAY,QAAS,KAARB,IAAb,C;QACb,MAAO,gBAAO,SAAP,C;QACP,MAAO,gBAAO,QAAP,C;  
QACP,OAAO,M;;QAEp,eAAa,iBAAa,SAAb,C;QACN,OAAP,QAAO,EAAO,QAAP,C;QACP,OAAO,Q;;K;IAIf,q  
C;MAII,aAAa,gB;MACN,OAAP,MAAO,EAAO,SAAP,C;MACA,SAAP,MAAO,EAAO,QAAP,C;MACP,OAAO,  
M;K;IAGX,qC;MAII,aAAa,iBAAa,SAAK,KAAL,GAAY,EAAZ,IAAb,C;MACb,MAAO,gBAAO,SAAP,C;MACA  
,SAAP,MAAO,EAAO,QAAP,C;MACP,OAAO,M;K;4FAGX,yB;MAAA,4C;MAAA,qC;QAKI,OAAO,gBAAK,O  
AAL,C;O;KALX,C;8FAQA,yB;MAAA,4C;MAAA,qC;QAKI,OAAO,gBAAK,OAAL,C;O;KALX,C;IAQA,yD;M  
AgB+C,oB;QAAA,OAAy,C;MAAG,8B;QAAA,iBAA0B,K;MAOzE,Q;MANX,oBAAoB,IAApB,EAA0B,IAA1B,  
C;MACA,IAAI,0CAAwB,8BAA5B,C;QACI,eAAe,SAAK,K;QACpB,qBAAqB,YAAW,IAAX,SAASB,YAAW,IA  
AX,UAAmB,CAAvB,GAA0B,CAA1B,GAAiC,CAAnD,K;QACrB,aAAa,iBAAmB,cAAAnB,C;QACb,gBAAy,CA  
AZ,C;QACA,Y;UAAO,c;UAAP,MAAgB,CAAT,mBAAiB,QAAxB,E;YAAA,K;UACI,iBAASB,eAAL,IAAK,EA  
Aa,WAAW,OAAx,IAAb,C;UACtB,IAAI,aAAa,IAAb,IAAqB,CAAC,cAA1B,C;YAA0C,K;UhBpnGID,WAAW,i  
BgBqnGa,UhBrnGb,C;UaCX,mBAAc,CAAd,YGonGwB,UHpnGxB,Y;YbA6B,egBonGS,sBHnnG3B,OGmnGgC,  
GAAK,OAAL,IAAL,ChBpnGT,C;;UgBonGrB,MAAO,WhBnnGR,IgBmnGQ,C;UACP,oBAAS,IAAT,I;;QAEJ,OA  
AO,M;;MAEX,eAAa,gB;MACiE,kBAA9E,iBAAiB,oBAAjB,EAA6B,IAA7B,EAAmC,IAAnC,EAAyC,cAAzC,E  
AAuE,KAAvE,C;MEpvGA,OAAGB,qBAAhB,C;QAAGB,gC;QFqvGL,mBErvGqB,OFqvGrB,C;;MAEX,OAAO,Q;  
K;IAGX,sE;MAKbKd,oB;QAAA,OAAy,C;MAAG,8B;QAAA,iBAA0B,K;MACvF,oBAAoB,IAApB,EAA0B,IAA  
1B,C;MACA,IAAI,0CAAwB,8BAA5B,C;QACI,eAAe,SAAK,K;QACpB,qBAAqB,YAAW,IAAX,SAASB,YAAW,  
IAAX,UAAmB,CAAvB,GAA0B,CAA1B,GAAiC,CAAnD,K;QACrB,aAAa,iBAAa,cAAAb,C;QACb,eAAa,kBAAc,

SAAd,C;QACb,YAAy,C;QACZ,OAAgB,CAAT,qBAAiB,QAAxB,C;UACI,iBAAsB,eAAL,IAAK,EAAa,WAAW, KAAX,IAAb,C;UACtB,IAAI,CAAC,cAAD,IAAmB,aAAa,IAApC,C;YAAOC,K;UAC1C,QAAO,cAAK,KAAL,E AAY,QAAQ,UAR,IAAZ,C;UACP,MAAO,WAAI,UAAU,QAAV,CAAJ,C;UACP,gBAAS,IAAT,I;QAEJ,OAA O,M;;MAEX,eAAa,gB;MACgE,kBAA7E,iBAAiB,oBAAjB,EAA6B,IAA7B,EAAmC,IAAnC,EAAyC,cAAzC,EA AuE,IAAvE,C;ME9xGA,OAAgB,qBAAhB,C;QAAgB,gC;QF+xGL,mBAAI,UE/xGiB,OF+xGjB,CAAJ,C;;MAEX, OAAO,Q;K;IAGX,kC;MAqBoB,gB;MAHhB,gBAXW,KAWW,O;MACtB,WAAW,iBFllGJ,MAAO,KEklGgB,mC AAwB,EAAxB,CFlIGhB,EEklG6C,SFlIG7C,CEklGH,C;MACX,QAAQ,C;MACQ,2B;MAAhB,OAAgB,cAAhB,C; QAAgB,yB;QACZ,IAAI,KAAC,SAAT,C;UAAoB,K;QACpB,IAAK,WAhBqB,GAgBP,OAhBO,EAAhB,KAgBqB ,CAAM,UAAN,EAAM,kBAAN,SAhBF,CAGrB,C;;MAhBT,OAKBO,I;K;+EafX,yB;MAAA,kF;MAAA,gE;MF/k GA,iB;ME+kGA,8C;QAWoB,UAEsB,M;QALtC,gBAAgB,KAAM,O;QACtB,WAAW,eFlIGJ,MAAO,KEklGgB,m CAAwB,EAAxB,CFlIGhB,EEklG6C,SFlIG7C,CEklGH,C;QACX,QAAQ,C;QACQ,2B;QAAhB,OAAgB,cAAhB,C ;UAAgB,yB;UACZ,IAAI,KAAC,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,OAAV,EAAmB,MAAM,UA AN,EAAM,kBAAN,SAAnB,CAAJ,C;;QAET,OAAO,I;O;KafX,C;IAkBA,kC;MAkBI,YAAy,oB;MACZ,aAZW,K AYQ,W;MACnB,WAAW,iBF/mGJ,MAAO,KE+mGgB,mCAAwB,EAAxB,CF/mGhB,EE+mGmD,wBAbtD,KAAs D,EAAwB,EAAxB,CF/mGnD,CE+mGH,C;MACX,OAAO,KAAM,UAAN,IAAmB,MAAO,UAAjC,C;QACI,IAA K,WafqB,GAeP,KAAM,OfC,EAeO,MAAO,Ofd,CAerB,C;;MAfT,OAIBo,I;K;+EAdX,yB;MAAA,kF;MAAA, gE;MF3mGA,iB;ME2mGA,8C;QAQI,YAAy,oB;QACZ,aAAa,KAAM,W;QACnB,WAAW,eF/mGJ,MAAO,KE+ mGgB,mCAAwB,EAAxB,CF/mGhB,EE+mGmD,wBAAN,KAAM,EAAwB,EAAxB,CF/mGnD,CE+mGH,C;QAC X,OAAO,KAAM,UAAN,IAAmB,MAAO,UAAjC,C;UACI,IAAK,WAAI,UAAU,KAAM,OAAhB,EAAwB,MAA O,OAA/B,CAAJ,C;;QAET,OAAO,I;O;KAdX,C;IAiBA,gC;MASW,sB;;QAaP,eAAe,oB;QACf,IAAI,CAAC,QAAS ,UAAAd,C;UAAyB,qBAAO,W;UAAP,uB;;QACzB,ahBvzGoD,gB;QgBwzGpD,cAAc,QAAS,O;QACvB,OAAO,QA AS,UAAhB,C;UACI,WAAW,QAAS,O;UACpB,MAAO,WAnBkB,GAmBJ,OAnBI,EAmBK,IAnBL,CAmBIB,C;U ACP,UAAU,I;QAEd,qBAAO,M;;MAtBP,yB;K;8FAGJ,yB;MAAA,qD;MhBjzGA,+D;MgBizGA,uC;QAUI,eAAe ,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,W;QACH,ahBvzGoD,gB;QgBwzGpD,cAAc,QAAS,O; QACvB,OAAO,QAAS,UAAhB,C;UACI,WAAW,QAAS,O;UACpB,MAAO,WAAI,UAAU,OAAV,EAAmB,IAAn B,CAAJ,C;UACP,UAAU,I;QAEd,OAAO,M;O;KAnBX,C;IASBA,8F;MAQ6D,yB;QAAA,YAA0B,I;MAAM,sB;Q AAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB ;QAAA,YAAoC,I;MAGtN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAy,C;MACI,2B;MAAhB,OAAgB,cA AhB,C;QAAgB,yB;QACZ,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR, IAAa,SAAS,KAA1B,C;UACW,gBAAP,MAAO,EAAC,OAAd,EAAuB,SAAvB,C;;UACJ,K;;MAEX,IAAI,SAAS,C AAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAA O,M;K;IAGX,4F;MAQwC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAA I,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;MACjN,OAAO,oBAAO,sBAAP, EAAwB,SAAxB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAAPD,EAA2D,SAAS3D,EAASe,SAATe,CAAI,F,W; K;4FAG5F,qB;MAKI,OAAO,S;K;IASS,8C;MAAA,mB;QAAE,OAAA,eAAK,W;O;K;IAN3B,iC;MAMI,oCAAgB ,8BAAhB,C;K;IAGJ,+B;MAOoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB, C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAhB,E;;MAEJ,OAAW,UAAS,CAAb,GA AgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MAOoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD, 2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAhB,E;;MA EJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MAOoB,Q;MAFhB,UAAkB, G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBA AnB,EAAmB,KAAhB,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B ;MAOoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OA AO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAhB,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GA AgC,MAAM,K;K;IAGjD,+B;MAOoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAA hB,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAhB,E;;MAEJ,OAAW,UAAS,CAAb, GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MAOoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MA CD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAhB,E;;

MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,2B;MAMoB,Q;MADhB,UA  
Ae,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAMo  
B,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,O;;MAEX,OAAO,G;K;I  
AGX,2B;MAMoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,OAAP,  
I;;MAEJ,OAAO,G;K;IAGX,2B;MAMoB,Q;MADhB,Y;MACgB,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAC  
Z,cAAO,OAAP,C;;MAEJ,OAAO,G;K;IAGX,2B;MAMoB,Q;MADhB,UAAiB,G;MACD,2B;MAAhB,OAAgB,cA  
AhB,C;QAAgB,yB;QACZ,OAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAMoB,Q;MADhB,UAAkB,G;MACF,2B;M  
AAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;;MAEX,OAAO,G;K;IG3+GX,uC;MAOI,OAAO,SAAM,C  
AAN,EAAS,SAAM,CAAN,EAAS,CAAT,EAAY,UAAZ,CAAT,EAakC,UAAIC,C;K;IAGX,oC;MAOI,OAAW,U  
AAW,SAAQ,CAAR,EAAS,CAAX,CAAX,IAA4B,CAAhC,GAAmC,CAAnC,GAA0C,C;K;IAmDrD,wC;MAQc,  
Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,IAAI,UAAW,SAAQ,GAAR,E  
AAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAC3D,OAAO,G;K;IA+GX,uC;MAOI,OAAO,SAAM  
,CAAN,EAAS,SAAM,CAAN,EAAS,CAAT,EAAY,UAAZ,CAAT,EAakC,UAAIC,C;K;IAGX,oC;MAOI,OAAW,  
UAAW,SAAQ,CAAR,EAAS,CAAX,CAAX,IAA4B,CAAhC,GAAmC,CAAnC,GAA0C,C;K;IAmDrD,wC;MAQc  
,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,IAAI,UAAW,SAAQ,GAAR,E  
AAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAC3D,OAAO,G;K;oGCnXX,yB;MAAA,iE;MAAA,  
uC;QASW,Q;QAAA,+B;;UAYS,U;UAAA,SnB8UoE,iBAAQ,W;UmB9U5F,OAAgB,gBAAhB,C;YAAgB,2B;YA  
CZ,aAbwB,SaAX,CAAU,OAAV,C;YACb,IAAI,cAAJ,C;cACI,8BAAO,M;cAAP,gC;;UAGR,8BAAO,I;;QALBA,  
kC;QAAA,iB;UAAmC,MAAM,gCAAuB,4DAAvB,C;;QAAhD,OAAO,I;O;KATX,C;gHAYA,gC;MASoB,Q;MA  
AA,OAAA,SnB8UoE,QAAQ,W;MmB9U5F,OAAgB,cAAhB,C;QAAgB,yB;QACZ,aAAa,UAAU,OAAV,C;QACb  
,IAAI,cAAJ,C;UACI,OAAO,M;;;MAGf,OAAO,I;K;IAGX,6B;MAIL,IAAI,mBAAQ,CAAZ,C;QACI,OAAO,W;M  
ACX,eAAe,iBAAQ,W;MACvB,IAAI,CAAC,QAAS,UAAAd,C;QACI,OAAO,W;MACX,YAAY,QAAS,O;MACrB,I  
AAI,CAAC,QAAS,UAAAd,C;QACI,OAAO,OnBgQiD,SmBhQ1C,KnBgQ+C,IAAL,EmBhQ1C,KnBgQoD,MAAV,  
CmBhQjD,C;;MACX,aAAa,iBAAsB,cAAtB,C;MACb,MAAO,WnB8PqD,SmB9PjD,KnB8PsD,IAAL,EmB9PjD,K  
nB8P2D,MAAV,CmB9PrD,C;;QAEwB,kBAAhB,QAAS,O;QAAPB,MAAO,WnB4PiD,SAAK,eAAL,EAAU,iBA  
AV,CmB5PjD,C;;MACO,QAAT,QAAS,W;MACIB,OAAO,M;K;uFAGX,yB;MAAA,+D;MASBA,gD;MatBA,uC;  
QAMW,kBAAU,gB;QAoBD,Q;QAAA,OnBuRoE,iBAAQ,W;QmBvR5F,OAAgB,cAAhB,C;UAAgB,yB;UACZ,W  
ArB6B,SAqBIB,CAAU,OAAV,C;UACC,OAAZ,WAAAY,EAAO,IAAP,C;;QAtBhB,OAwbO,W;O;KA9BX,C;uFA  
SA,yB;MAAA,+D;MAwBA,gD;MAxBA,uC;QAUW,kBAAU,gB;QAsBD,Q;QAAA,OnBwQoE,iBAAQ,W;QmBx  
Q5F,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAvB6B,SAuBIB,CAAU,OAAV,C;UACC,OAAZ,WAAAY,EAAO,IA  
AP,C;;QAxBhB,OA0BO,W;O;KApCX,C;2FAaA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAA,OAAA,SnBuRoE,  
QAAQ,W;QmBvR5F,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAAY,EA  
AO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;2FAWA,yB;MAAA,gD;MAAA,oD;QAQoB,Q;QAAA,OAAA,SnBw  
QoE,QAAQ,W;QmBxQ5F,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAA  
Y,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAZX,C;8EAeA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,cAA  
b,C;QA2BA,Q;QAAA,OnB+NuE,iBAAQ,W;QmB/N5F,OAAa,cAAb,C;UAAa,sB;UACT,WAAAY,WA5BiB,SA4B  
b,CAAU,IAAV,CAAJ,C;;QA5BhB,OA6BO,W;O;KApCX,C;4FAUA,yB;MAAA,+D;MAAA,uC;QAOW,kBAaA,g  
B;QA4EJ,Q;QAAA,OnBoKoE,iBAAQ,W;QmBpK5F,OAAgB,cAAhB,C;UAAgB,yB;UApEK,U;UAAA,cARe,SA  
Qf,CAoEQ,OApER,W;YAAc,6B;;;QAR3D,OASO,W;O;KAhBX,C;gGAUA,yB;MAAA,oD;QAYeOB,Q;QAAA,  
OnBoKoE,iBAAQ,W;QmBpK5F,OAAgB,cAAhB,C;UAAgB,yB;UApEK,U;UAAA,wBAoEQ,OApER,W;YAAc,  
6B;;;QAC3D,OAAO,W;O;KANX,C;kFASA,6C;MAKiB,Q;MAAA,OAAA,SnB+NuE,QAAQ,W;MmB/N5F,OAAa  
,cAAb,C;QAAa,sB;QACT,WAAAY,WAAL,UAAU,IAAV,CAAJ,C;;MACHB,OAAO,W;K;8EAGX,gC;MAOoB,Q;  
MADhB,IAAI,mBAAJ,C;QAae,OAAO,I;MACN,OAAA,SnBmNoE,QAAQ,W;MmBnN5F,OAAgB,cAAhB,C;QA  
AgB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,OAAO,K;;MACTd,OAAO,I;K;IAGX,2B;MAMI,O  
AAO,CAAC,mB;K;+EAGZ,gC;MAOoB,Q;MADhB,IAAI,mBAAJ,C;QAae,OAAO,K;MACN,OAAA,SnB+LoE,Q  
AAQ,W;MmB/L5F,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,I;;MAC  
rD,OAAO,K;K;mFAGX,qB;MAKI,OAAO,c;K;mFAGX,gC;MAMoB,Q;MAFhB,IAAI,mBAAJ,C;QAae,OAAO,C  
;MACtB,YAAY,C;MACI,OAAA,SnB6KoE,QAAQ,W;MmB7K5F,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,U

AAU,OAAV,CAAJ,C;UAAwB,qB;;MAC9C,OAAO,K;K;sFAGX,6B;MAKOB,Q;MAAA,OAAA,SnBoKoE,QAA Q,W;MmBpK5F,OAAGB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;K;kFAG1B,yB;MJ+qDA,sE;MI/qDA,s C;QAYmB,kBAAR,iB;QAAQ,gB;;UJ8qDf,eAAe,sB;UACf,IAAI,CAAC,QAAS,UAAAd,C;YAAyB,MAAM,6B;UA C/B,cAAc,QAAS,O;UACvB,IAAI,CAAC,QAAS,UAAAd,C;YAAyB,eAAO,O;YAAP,iB;;UACzB,eIrrDqB,QJkrDN, CAAS,OAAT,C;;YAEX,QAAQ,QAAS,O;YACjB,QIrrDiB,QJqrDT,CAAS,CAAT,C;YACR,IAAI,2BAAW,CAAX ,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;;UAED,QAAT,QAAS,W;UACIB,eAAO,O;;;QI3rDP,mB;O;KAZJ,C;8 FAeA,+B;MAQmB,kBAAR,iB;MAAQ,sB;;QJ0rDf,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAA O,I;UAAP,uB;;QACzB,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,O;UAAP,uB;;QA CzB,eI9rD2B,QJ8rDZ,CAAS,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QIjsDuB,QJisDf,CAAS,CAAT,C;UACR, IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAED,QAAT,QAAS,W;QACIB,qBAAO,O;;; MIvsDP,yB;K;mFAGJ,yB;MJusDA,sE;MF/2DA,iB;MMwKA,sC;QJotDI,eIvsDO,iBJusDQ,W;QACf,IAAI,CAAC, QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eIzsDqB,QJysDN,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UA AhB,C;UACI,QI3sDiB,QJ2sDT,CAAS,QAAS,OAAIB,C;UACR,WFz3DG,MAAO,KEy3DO,QFz3DP,EEy3DiB,C Fz3DjB,C;;QM6Kd,OJ8sDO,Q;O;KI3tDX,C;mFagBA,yB;MJ8sDA,sE;MFj5DA,iB;MMmMA,sC;QJ2tDI,eI9sDO, iBJ8sDQ,W;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eIhtDqB,QJgtDN,CAAS,QAAS,OA AIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIltDiB,QJktDT,CAAS,QAAS,OAAIB,C;UACR,WF35DG,MAAO, KE25DO,QF35DP,EE25DiB,CF35DjB,C;;QMwMd,OJqtDO,Q;O;KIluDX,C;mFagBA,yB;MJqtDA,sE;MIrtDA,sC ;QJguDI,eIrtDO,iBJqtDQ,W;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eIvtDqB,QJutDN,C AAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIztDiB,QJytDT,CAAS,QAAS,OAAIB,C;UACR,I AAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QI3tDnB,OJ8tDO,Q;O;KIzuDX,C;+FACa,yB;MNTNA,iB;MMs NA,sC;QAWmB,kBAAR,iB;QAAQ,sB;;UJ8tDf,eAAe,sB;UACf,IAAI,CAAC,QAAS,UAAAd,C;YAAyB,qBAAO,I; YAAP,uB;;UACzB,eIhuD2B,QJguDZ,CAAS,QAAS,OAAIB,C;UACf,OAAO,QAAS,UAAhB,C;YACI,QIluDuB,Q JkuDf,CAAS,QAAS,OAAIB,C;YACR,WF57DG,MAAO,KE47DO,QF57DP,EE47DiB,CF57DjB,C;;UE87Dd,qBA AO,Q;;QIruDP,yB;O;KAXJ,C;+FACa,yB;MN/OA,iB;MM+OA,sC;QAWmB,kBAAR,iB;QAAQ,sB;;UJquDf,eAA e,sB;UACf,IAAI,CAAC,QAAS,UAAAd,C;YAAyB,qBAAO,I;YAAP,uB;;UACzB,eIvuD2B,QJuuDZ,CAAS,QAAS, OAAIB,C;UACf,OAAO,QAAS,UAAhB,C;YACI,QIzuDuB,QJyuDf,CAAS,QAAS,OAAIB,C;YACR,WF59DG,M AAO,KE49DO,QF59DP,EE49DiB,CF59DjB,C;;UE89Dd,qBAAO,Q;;;QI5uDP,yB;O;KAXJ,C;+FACa,+B;MASm B,kBAAR,iB;MAAQ,sB;;QJ4uDf,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,I;UAAP,uB;;QA CzB,eI9uD2B,QJ8uDZ,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIhvDuB,QJgvDf,CAAS, QAAS,OAAIB,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,qBAAO,Q;;;MIrvDP,yB;K;0 FAGJ,yB;MJqvDA,sE;MIrvDA,kD;QJgwDI,eIrvDO,iBJqvDQ,W;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,M AAM,6B;QAC/B,eIvDqC,QJuvDtB,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIzvDiC,QJ yvDzB,CAAS,QAAS,OAAIB,C;UACR,II1vDqB,UJ0vDN,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAATC, C;YACI,WAAW,C;;;QI3vDnB,OJ8vDO,Q;O;KIzwDX,C;sGAcA,2C;MASmB,kBAAR,iB;MAAQ,0B;;QJ8vDf,eA Ae,sB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,yBAAO,I;UAAP,2B;;QACzB,eIhwD2C,QJgwD5B,CAAS,QA AS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIiwDuC,QJkwD/B,CAAS,QAAS,OAAIB,C;UACR,IIhwD2 B,UJmwDZ,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAATC,C;YACI,WAAW,C;;;QAGnB,yBAAO,Q;;;MI vwDP,6B;K;sFAGJ,yB;MAAA,kD;MAAA,wC;QAUI,OAAe,QAAR,iBAAQ,EAAQ,UAAR,C;O;KAVnB,C;kGAa A,yB;MAAA,8D;MAAA,wC;QAMI,OAAe,cAAR,iBAAQ,EAAc,UAAAd,C;O;KANnB,C;kFASA,yB;MJi4DA,sE; MIj4DA,sC;QAYmB,kBAAR,iB;QAAQ,gB;;UJg4Df,eAAe,sB;UACf,IAAI,CAAC,QAAS,UAAAd,C;YAAyB,MAA M,6B;UAC/B,cAAc,QAAS,O;UACvB,IAAI,CAAC,QAAS,UAAAd,C;YAAyB,eAAO,O;YAAP,iB;;UACzB,eIp4Dq B,QJo4DN,CAAS,OAAT,C;;YAEX,QAAQ,QAAS,O;YACjB,QIv4DiB,QJu4DT,CAAS,CAAT,C;YACR,IAAI,2B AAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;;UAED,QAAT,QAAS,W;UACIB,eAAO,O;;;QI74DP,m B;O;KAZJ,C;8FAeA,+B;MAQmB,kBAAR,iB;MAAQ,sB;;QJ44Df,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAAd,C; UAAyB,qBAAO,I;UAAP,uB;;QACzB,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,O; UAAP,uB;;QACzB,eIh5D2B,QJg5DZ,CAAS,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QIn5DuB,QJm5Df,CAA S,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAED,QAAT,QAAS,W;Q ACIB,qBAAO,O;;;MIz5DP,yB;K;mFAGJ,yB;MJy5DA,sE;MF7gEA,iB;MMoHA,sC;QJs6DI,eIz5DO,iBJy5DQ,W;

QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eI35DqB,QJ25DN,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QI75DiB,QJ65DT,CAAS,QAAS,OAAIB,C;UACR,WFvhEG,MAAO,KEuhEO,QFvhEP,EEuhEiB,CFvhEjB,C;;QMyHd,OJg6DO,Q;O;KI76DX,C;mFAGBA,yB;MJg6DA,sE;MF/iEA,iB;MM+IA,sC;QJ66DI,eIh6DO,iBJg6DQ,W;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eI16DqB,QJk6DN,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIp6DiB,QJo6DT,CAAS,QAAS,OAAIB,C;UACR,WFzjEG,MAAO,KEyjEO,QFzjEP,EEyjEiB,CFzjEjB,C;;QMoJd,OJu6DO,Q;O;KI77DX,C;mFAGBA,yB;MJu6DA,sE;MIv6DA,sC;QJk7DI,eIv6DO,iBJu6DQ,W;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eIz6DqB,QJy6DN,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QI36DiB,QJ26DT,CAAS,QAAS,OAAIB,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,WAAW,C;;;QI76DnB,OJg7DO,Q;O;KI37DX,C;+FACa,yB;MNIKA,iB;MMkKA,sC;QAWmB,kBAAR,iB;QAAQ,sB;;UJg7Df,eAAe,sB;UACf,IAAI,CAAC,QAAS,UAAAd,C;YAAyB,qBAAO,I;YAAP,uB;;UACzB,eI17D2B,QJk7DZ,CAAS,QAAS,OAAIB,C;UACf,OAAO,QAAS,UAAhB,C;YACI,QIp7DuB,QJo7Df,CAAS,QAAS,OAAIB,C;YACR,WF11EG,MAAO,KE01EO,QF11EP,EE01EiB,CF11EjB,C;;UE4IEd,qBAAO,Q;;;QIv7DP,yB;O;KAXJ,C;+FACa,yB;MN3LA,iB;MM2LA,sC;QAWmB,kBAAR,iB;QAAQ,sB;;UJu7Df,eAAe,sB;UACf,IAAI,CAAC,QAAS,UAAAd,C;YAAyB,qBAAO,I;YAAP,uB;;UACzB,eIz7D2B,QJy7DZ,CAAS,QAAS,OAAIB,C;UACf,OAAO,QAAS,UAAhB,C;YACI,QI37DuB,QJ27Df,CAAS,QAAS,OAAIB,C;YACR,WF1nEG,MAAO,KE0nEO,QF1nEP,EE0nEiB,CF1nEjB,C;;UE4nEd,qBAAO,Q;;;QI97DP,yB;O;KAXJ,C;+FACa,+B;MASmB,kBAAR,iB;MAAQ,sB;;QJ87Df,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,I;YAAP,uB;;QACzB,eIh8D2B,QJg8DZ,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QI18DuB,QJk8Df,CAAS,QAAS,OAAIB,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,WAAW,C;;;QAGnB,qBAAO,Q;;;MIv8DP,yB;K;0FAGJ,yB;MJu8DA,sE;MIv8DA,kD;QJk9DI,eIv8DO,iBJu8DQ,W;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eIz8DqC,QJy8DtB,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QI38DiC,QJ28DzB,CAAS,QAAS,OAAIB,C;UACR,II58DqB,UJ48DN,SAAQ,QAAR,EAakB,CAAIB,C;AAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QI78DnB,OJg9DO,Q;O;KI39DX,C;sGAcA,2C;MASmB,kBAAR,iB;MAAQ,0B;;QJg9Df,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,yBAAO,I;YAAP,2B;;QACzB,eI19D2C,QJk9D5B,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIp9DuC,QJo9D/B,CAAS,QAAS,OAAIB,C;UACR,I1r9D2B,UJq9DZ,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,yBAAO,Q;;;MIz9DP,6B;K;sFAGJ,yB;MAAA,kD;MAAA,wC;QAUI,OAAe,QAAR,iBAAQ,EAAQ,UAAR,C;O;KAVnB,C;kGAaA,yB;MAAA,8D;MAAA,wC;QAMI,OAAe,cAAR,iBAAQ,EAAc,UAAAd,C;O;KANnB,C;IASA,4B;MAMI,OAAO,mB;K;iFAGX,gC;MAOoB,Q;MADhB,IAAI,mBAAJ,C;QAae,OAAO,I;MACN,OAAA,SnB/KoE,QAAQ,W;MmB+K5F,OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;MACrD,OAAO,I;K;oFAGX,6B;MAKmC,Q;MAAA,OnBxLqD,iBAAQ,W;MmBwL7E,OAAgB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;MAArC,gB;K;kGAGJ,yB;MAAA,6B;MAAA,sC;MJ4wCA,wE;MI5wCA,2BAQiB,yB;QJowCjB,wE;elpwCiB,0B;UAAA,4B;YAAU,kBAAR,iB;YAAQ,aAAe,c;YJ2wCzB,gB;YADb,YAAY,C;YACC,6B;YAAb,OAAa,cAAb,C;cAAa,sB;cAAM,OAAO,oBAAmB,cAAnB,EAAMb,sBAAnB,UAAP,EAAoC,IAApC,C;;YI3wC2B,W;W;S;OAAjC,C;MARjB,oC;QJmxCiB,gB;QADb,YAAY,C;QACC,OI3wCE,iBJ2wCF,W;QAAb,OAAa,cAAb,C;UAAA,sB;UAAM,OAAO,oBAAmB,cAAnB,EAAMb,sBAAnB,UAAP,EAAoC,IAApC,C;;QI3wCnB,gB;O;KARJ,C;4FAWA,qB;MAKI,OAAO,iB;K;IAGX,iC;MAIL,OAAe,aAAR,iBAAQ,C;K;IC5jBnB,kC;MAEI,gBCmE2D,8BAAy,c;MDIEvE,IAAI,SAAU,OAAV,GAAMb,CAAvB,C;QACW,Q;QAAA,IAAI,cAAQ,GAAZ,C;UAAA,OAAsB,S;;uBAAe,qBAAU,CAAV,C;UAAA,YAAe,SE0Oc,WF1OM,CE0ON,CAXcF,c;UFIMnD,OG8MoD,2BAAL,GAakB,K;;QH9MxE,W;;MAEJ,OAAuB,oBAAhB,wBAAGB,C;K;IzBD3B,6B;MAOI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAc,SAAAd,eAAvB,C;MACV,OAAO,SAAK,M;K;IAGhB,6B;MAOI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAc,SAAAd,eAAvB,C;MACV,OAAO,SAAK,M;K;IAGhB,6B;MAOI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAc,SAAAd,eAAvB,C;MACV,OAAO,SAAK,M;K;IAGhB,mC;MAKI,OAAW,mBAAJ,GA Ae,IAAf,GAAYB,SAAK,M;K;IAGzC,mC;MAKI,OAAW,mBAAJ,GA Ae,IAAf,GAAYB,SAAK,M;K;IAGzC,mC;MAKI,OAAW,mBAAJ,GA Ae,IAAf,GAAYB,SAAK,M;K;IAGzC,4B;MASI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAc,SAAAd,eAAvB,C;MACV,OAAO,SAAK,K;K;IAGhB,4B;MASI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAc,SAAAd,eAAvB,C;MACV,OAAO,SAAK,K;K;IAGhB,4B;MASI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAc,SAAAd,eAAvB,C;MACV,OAAO,SAAK,K;K;IAGhB,kC;MAOI,OAAW,mBAAJ,GA Ae,IAAf,GAAYB,SAA

K,K;K;IAGzC,kC;MAOI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,SAAK,K;K;IAGzC,kC;MAOI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,SAAK,K;K;gFAGzC,yB;MAAA,mC;MAAA,2C;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;gFAWA,yB;MAAA,mC;MAAA,2C;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;gFAWA,yB;MAAA,mC;MAAA,2C;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;IAWA,sC;;QAQQ,OAAc,QAAP,MAAO,EAAQ,SAAR,C;;QACHB,+C;UACE,MAAM,2BAAuB,CAAe,QAAzB,C;;UAHV,O;;K;IAOJ,sC;;QAQQ,OAAc,SAAP,MAAO,EAAS,SAAT,C;;QACHB,+C;UACE,MAAM,2BAAuB,CAAe,QAAzB,C;;UAHV,O;;K;IAOJ,sC;;QAQQ,OAAiD,OAA1C,MAAO,iBAAQ,e6BzKgB,I7ByKxB,EAAoB,CAAA,c6BzKI,I7ByKJ,IAAY,CAAZ,IAApB,CAAmC,C;;QACnD,+C;UACE,MAAM,2BAAuB,CAAe,QAAzB,C;;UAHV,O;;K;4FAOJ,yB;MAAA,mC;MAAA,uD;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;4FAUA,yB;MAAA,mC;MAAA,uD;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;4FAUA,yB;MAAA,mC;MAAA,uD;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;IAUA,4C;MAMI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAc,QAAP,MAAO,EAAQ,SAAR,C;K;IAGIB,4C;MAMI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAc,SAAP,MAAO,EAAS,SAAT,C;K;IAGIB,4C;MAMI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAiD,OAA1C,MAAO,iBAAQ,e6B3OoB,I7B2O5B,EA AoB,CAAA,c6B3OQ,I7B2OR,IAAY,CAAZ,IAApB,CAAmC,C;K;mFAGrD,8B;MAQI,OAAO,mBAAmB,2BAAS,OAAT,C;K;oFAG9B,8B;MAQI,OAAO,mBAAmB,2BAAS,OAAT,C;K;oFAG9B,8B;MAQI,OAAO,mBAAmB,2BAAS,OAAT,C;K;IAG9B,uC;MAKI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAKI,OAAO,2BAAe,oBAAN,KAAM,CAAf,C;K;IAGX,uC;MAKI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAOI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAOI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAOI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAOI,OAAO,2BAAe,oBAAN,KAAM,CAAf,C;K;IAGX,uC;MAOI,OAAO,2BAAe,KAAf,C;K;oFAGX,yB;MAAA,6C;MAAA,8B;MAAA,+C;MAAA,mC;QAKY,Q;QAAR,OAAkC,SAA1B,gEAA0B,EAAS,KAAT,C;O;KALtC,C;oFAQA,yB;MAAA,6C;MAAA,8B;MAAA,+C;MAAA,mC;QAKY,Q;QAAR,OAAmC,SAA3B,gEAA2B,EAAS,KAAT,C;O;KALvC,C;IAQA,uC;MiB3SW,SjBkTM,mBAAN,KAAM,C;MAAb,OAA0C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG5E,uC;MiBrTW,SjB4TM,kBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG7E,uC;MiB/TW,SjBsUM,oBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG7E,uC;MiBzUW,SjBgVM,qBAAN,KAAM,C;MAAb,OAA4C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG9E,uC;MAKI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MiB3VW,SjBkWM,mBAAN,KAAM,C;MAAb,OAA0C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG5E,uC;MiBrWW,SjB4WM,oBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG7E,uC;MiB/WW,SjBsXM,oBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG7E,uC;MiBzXW,SjBgYM,qBAAN,KAAM,C;MAAb,OAA4C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG9E,uC;MAKI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAKI,OAAO,2BAAe,oBAAN,KAAM,CAAf,C;K;IAGX,uC;MiB7ZW,SjBkaM,kBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG7E,uC;MiBraW,SjB0aM,mBAAN,KAAM,C;MAAb,OAA4C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG9E,uC;MAOI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAOI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAOI,OAAO,2BAAe,oBAAN,KAAM,CAAf,C;K;IAGX,uC;MiB3cW,SjBkdM,kBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG7E,uC;MiBrdW,SjB4dM,mBAAN,KAAM,C;MAAb,OAA4C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;oFAG9E,yB;MAAA,6C;MAAA,8B;MAAA,+C;MAAA,mC;QAKY,Q;QAAR,OAAmC,SAA3B,gEAA2B,EAAS,KAAT,C;O;KALvC,C;IAQA,uC;MiBveW,SjB4eM,iBAAN,KAAM,C;MAAb,OAA0C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG5E,uC;MiB/eW,SjBofM,oBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG7E,uC;MiBvfW,SjB4fM,qBAAN,KAAM,C;MAAb,OAA4C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG9E,uC;MAOI,OAAO,2BAAS,KAAM,WAAf,C;K;IAGX,uC;MAOI,OAAO,2BAAS,KAAM,WAAf,C;K;IAGX,uC;MiBnhBW,SjB0hBM,iBAAN,KAAM,C;MAAb,OAA0C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG5E,uC;MiB7hBW,SjBoiBM,oBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG7E,uC;MiBviBW,SjB8iBM,qBAAN,KAAM,C;MAAb,OAA4C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;oFAG9E,yB;MAAA,6C;MAAA,8B;MAAA,+C;MAAA,mC;QAKY,Q;QAAR,OAAkC,SAA1B,gEAA0B,EAAS,KAAT,C;O;KALtC,C;IAQA,uC;MAKI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAKI,OAAO,2BAAe,oBAAN,KAAM,CAAf,C;K;IAGX,uC;MiBzkBW,SjB8kBM,oBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EA





AA0C,mCAA1C,CAAR,4BAAJ,GAAuE,uBAAL,SAAK,CAAvE,GAAqF,I;K;IAGhG,wC;MACI,OAAmB,UAAe,  
mCAAf,EAAyC,mCAAzC,CAAR,4BAAJ,GAAqE,uBAAL,SAAK,CAArE,GAAmF,I;K;IAG9F,uC;MACI,OAAm  
B,MAAR,8BAAiC,KAArC,GAAmE,QAAL,SAAK,CAAnE,GAAkF,I;K;IAG7F,yC;MACI,OAAW,uEAAJ,GAAq  
E,QAAL,SAAK,SAArE,GAAoF,I;K;IAG/F,yC;MACI,OAAmB,UAAA,uCAAgB,UAAhB,EAA4B,uCAAgB,UAA  
5C,CAAR,4BAAJ,GAAiF,QAAR,YAAL,SAAK,CAAQ,CAAjF,GAAgG,I;K;IAG3G,yC;MACI,OAAmB,UAAA,u  
CAAgB,UAAhB,EAA2B,uCAAgB,UAA3C,CAAR,4BAAJ,GAA+E,QAAR,YAAL,SAAK,CAAQ,CAA/E,GAA8F  
,I;K;IAGzG,8B;MAMI,OAAO,wBAAy,EAAa,GAAH,CAAG,IAAzB,C;K;IAGX,gC;MAMI,OAAO,kBAAy,oBA  
AH,EAAG,CAAc,8BAAH,CAAG,EAA1B,C;K;IAGX,gC;MAMI,OAAO,aAAK,SAAL,EAAoB,EAAa,GAAH,CA  
AG,IAAjC,C;K;IAGX,gC;MAMI,OAAO,aAAK,SAAL,EAAoB,EAAa,GAAH,CAAG,IAAjC,C;K;IAGX,gC;MA  
MI,IAAI,MAAM,CAA V,C;QAAoB,OAAO,iCAAU,M;MACrC,OAAO,yBAAiB,OAAR,EAAQ,GAAH,CAAG,CA  
AjB,C;K;IAGX,gC;MAMI,IAAI,MAAM,WAAV,C;QAAyB,OAAO,gCAAS,M;MACzC,OAAO,wBAAS,EAAQ,G  
AAH,CAAG,IAAjB,C;K;IAGX,gC;MAMI,OAAO,kBAAy,oBAAH,EAAG,CAAc,8BAAH,CAAG,EAA1B,C;K;I  
AGX,gC;MAMI,IAAI,MAAM,WAAV,C;QAAyB,OAAO,gCAAS,M;MACzC,OAAO,aAAK,SAAL,EAAiB,EAA  
Q,GAAH,CAAG,IAAzB,C;K;IAGX,gC;MAMI,IAAI,MAAM,WAAV,C;QAAyB,OAAO,gCAAS,M;MACzC,OA  
AO,aAAK,SAAL,EAAiB,EAAQ,GAAH,CAAG,IAAzB,C;K;IAGX,gC;MAMI,IAAI,iDAAJ,C;QAA0B,OAAO,iC  
AAU,M;MAC3C,OAAy,oBAAL,SAAK,CAAL,SAAkB,EAAQ,8BAAH,CAAG,EAA1B,C;K;IAGX,gC;MAMI,IA  
AI,iDAAJ,C;QAA0B,OAAO,iCAAU,M;MAC3C,OAAO,kBAAS,EAAQ,8BAAH,CAAG,EAAjB,C;K;IAGX,iC;M  
AMI,IAAI,iDAAJ,C;QAA0B,OAAO,iCAAU,M;MAC3C,OAAy,oBAAL,SAAK,CAAL,SAAkB,EAAQ,8BAAH,C  
AAG,EAA1B,C;K;IAGX,iC;MAMI,IAAI,iDAAJ,C;QAA0B,OAAO,iCAAU,M;MAC3C,OAAy,oBAAL,SAAK,C  
AAL,SAAkB,EAAQ,8BAAH,CAAG,EAA1B,C;K;IAGX,iC;MAMI,OAAO,wBAAy,EAAa,GAAH,CAAG,IAAzB  
,C;K;IAGX,iC;MAMI,OAAO,kBAAy,oBAAH,EAAG,CAAc,8BAAH,CAAG,EAA1B,C;K;IAGX,iC;MAMI,OAA  
O,aAAK,SAAL,EAAoB,EAAa,GAAH,CAAG,IAAjC,C;K;IAGX,iC;MAMI,OAAO,aAAK,SAAL,EAAoB,EAAa,  
GAAH,CAAG,IAAjC,C;K;IAGX,gD;MAQI,OAAW,4BAAO,YAAP,KAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,  
kD;MAQI,OAAW,YAAO,YAAX,GAAyB,YAAzB,GAA2C,S;K;IAGtD,kD;MAQI,OAAW,YAAO,YAAX,GAAy  
B,YAAzB,GAA2C,S;K;IAGtD,kD;MAQI,OAAW,YAAO,YAAX,GAAyB,YAAzB,GAA2C,S;K;IAGtD,kD;MAQI  
,OAAW,0BAAO,YAAP,KAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,kD;MAQI,OAAW,YAAO,YAAX,GAAyB,  
YAAzB,GAA2C,S;K;IAGtD,kD;MAQI,OAAW,YAAO,YAAX,GAAyB,YAAzB,GAA2C,S;K;IAGtD,+C;MAQI,O  
AAW,4BAAO,YAAP,KAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAQI,OAAW,YAAO,YAAX,GAAyB,YA  
AzB,GAA2C,S;K;IAGtD,iD;MAQI,OAAW,YAAO,YAAX,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAQI,OAA  
W,YAAO,YAAX,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAQI,OAAW,0BAAO,YAAP,KAAJ,GAAyB,YAAzB  
,GAA2C,S;K;IAGtD,iD;MAQI,OAAW,YAAO,YAAX,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAQI,OAAW,Y  
AAO,YAAX,GAAyB,YAAzB,GAA2C,S;K;IAGtD,yD;MAQI,IAAI,iBAAiB,IAAjB,IAAyB,iBAAiB,IAA9C,C;Q  
ACI,IAAI,+BAAe,YAAf,KAAJ,C;UAAiC,MAAM,gCAAyB,6DAAiD,YAAjD,wCAAoF,YAApF,OAAzB,C;QAC  
vC,IAAI,4BAAO,YAAP,KAAJ,C;UAAyB,OAAO,Y;QAChC,IAAI,4BAAO,YAAP,KAAJ,C;UAAyB,OAAO,Y;Q  
AGhC,IAAI,iBAAiB,IAAjB,IAAyB,4BAAO,YAAP,KAA7B,C;UAAkD,OAAO,Y;QACzD,IAAI,iBAAiB,IAAjB,I  
AAyB,4BAAO,YAAP,KAA7B,C;UAAkD,OAAO,Y;MAE7D,OAAO,S;K;IAGX,2D;MAQI,IAAI,eAAe,YAAAnB,  
C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,8BAAoF,YAApF,MAAzB,C;MACvC,IAAI,YAAO,YAAX,C;QAA  
yB,OAAO,Y;MACHC,IAAI,YAAO,YAAX,C;QAAyB,OAAO,Y;MACHC,OAAO,S;K;IAGX,2D;MAQI,IAAI,eAA  
e,YAAAnB,C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,8BAAoF,YAApF,MAAzB,C;MACvC,IAAI,YAAO,YA  
AX,C;QAAyB,OAAO,Y;MACHC,IAAI,YAAO,YAAX,C;QAAyB,OAAO,Y;MACHC,OAAO,S;K;IAGX,2D;MAQ  
I,IAAI,eAAe,YAAAnB,C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,8BAAoF,YAApF,MAAzB,C;MACvC,IAAI,  
YAAO,YAAX,C;QAAyB,OAAO,Y;MACHC,IAAI,YAAO,YAAX,C;QAAyB,OAAO,Y;MACHC,OAAO,S;K;IAG  
X,2D;MAQI,IAAI,6BAAe,YAAf,KAAJ,C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,yCAAoF,YAApF,iBAAzB  
,C;MACvC,IAAI,0BAAO,YAAP,KAAJ,C;QAAyB,OAAO,Y;MACHC,IAAI,0BAAO,YAAP,KAAJ,C;QAAyB,OA  
AO,Y;MACHC,OAAO,S;K;IAGX,2D;MAQI,IAAI,eAAe,YAAAnB,C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,8  
BAAoF,YAApF,MAAzB,C;MACvC,IAAI,YAAO,YAAX,C;QAAyB,OAAO,Y;MACHC,IAAI,YAAO,YAAX,C;Q  
AAyB,OAAO,Y;MACHC,OAAO,S;K;IAGX,2D;MAQI,IAAI,eAAe,YAAAnB,C;QAAiC,MAAM,gCAAyB,oDAAi  
D,YAAjD,8BAAoF,YAApF,MAAzB,C;MACvC,IAAI,YAAO,YAAX,C;QAAyB,OAAO,Y;MACHC,IAAI,YAAO,

YAAX,C;QAAyB,OAAO,Y;MACHC,OAAO,S;K;IAGX,sC;MAUW,Q;MADP,IAAI,KAAM,UAAV,C;QAAqB,M  
AAM,gCAAyB,4CAAyC,KAAzC,MAAzB,C;MAGvB,IAAA,KAAM,0BAAiB,SAAjB,EAAuB,KAAM,MAA7B,  
CAAN,IAA6C,CAAC,KAAM,0BAAiB,KAAM,MAAvB,EAA8B,SAA9B,CAApD,C;QAAiG,OAAN,KAAM,M;  
WAEjG,IAAA,KAAM,0BAAiB,KAAM,aAAvB,EAAqC,SAArC,CAAN,IAAoD,CAAC,KAAM,0BAAiB,SAAjB,  
EAAuB,KAAM,aAA7B,CAA3D,C;QAA+G,OAAN,KAAM,a;;QACvG,gB;MALZ,W;K;IASJ,sC;MAYW,Q;MAJP  
,IAAI,8CAAJ,C;QACI,OAAy,WAAL,SAAK,EAAy,KAAZ,C;;MAEhB,IAAI,KAAM,UAAV,C;QAAqB,MAAM,  
gCAAyB,4CAAyC,KAAzC,MAAzB,C;MAEvB,gCAAO,KAAM,MAAb,M;QAA4B,OAAN,KAAM,M;WAC5B,g  
CAAO,KAAM,aAAb,M;QAAMC,OAAN,KAAM,a;;QAC3B,gB;MAHZ,W;K;IAOJ,sC;MAYW,Q;MAJP,IAAI,8C  
AAJ,C;QACI,OAAy,WAAL,SAAK,EAAc,KAAc,C;;MAEhB,IAAI,KAAM,UAAV,C;QAAqB,MAAM,gCAAyB,  
4CAAyC,KAAzC,MAAzB,C;MAEvB,gBAAO,KAAM,MAAb,C;QAA4B,OAAN,KAAM,M;WAC5B,gBAAO,KA  
AM,aAAb,C;QAAMC,OAAN,KAAM,a;;QAC3B,gB;MAHZ,W;K;IAOJ,sC;MAYW,Q;MAJP,IAAI,8CAAJ,C;QA  
CI,OAAy,WAAL,SAAK,EAAe,KAAf,C;;MAEhB,IAAI,KAAM,UAAV,C;QAAqB,MAAM,gCAAyB,4CAAyC,K  
AAzC,MAAzB,C;MAEvB,8BAAO,KAAM,MAAb,M;QAA4B,OAAN,KAAM,M;WAC5B,8BAAO,KAAM,aAAb,  
M;QAAMC,OAAN,KAAM,a;;QAC3B,gB;MAHZ,W;K;IY5rDJ,oD;MAMuF,wC;K;IANvF,8CAOI,Y;MAAuC,8B;  
K;IAP3C,gF;IkBQA,yC;MAMI,OAAO,sBAAQ,OAAR,KAAoB,C;K;IAWG,2C;MAAA,qB;QAAE,MAAM,8BAA  
0B,+CAA4C,aAA5C,MAA1B,C;O;K;IAR1C,uC;MAQI,OAAO,8BAAgB,KAAhB,EAAuB,yBAAvB,C;K;IAGX,4  
D;MAcQb,Q;MANjB,IAAI,QAAQ,CAAZ,C;QACI,OAAO,aAAa,KAAb,C;MACX,eAAe,oB;MACf,YAAy,C;MA  
CZ,OAAO,QAAS,UAAhB,C;QACI,cAAc,QAAS,O;QACvB,IAAI,WAAS,YAAT,EAAS,oBAAT,OAAJ,C;UACI,  
OAAO,O;;MAEf,OAAO,aAAa,KAAb,C;K;IAGX,8C;MAcQb,Q;MANjB,IAAI,QAAQ,CAAZ,C;QACI,OAAO,I;M  
ACX,eAAe,oB;MACf,YAAy,C;MACZ,OAAO,QAAS,UAAhB,C;QACI,cAAc,QAAS,O;QACvB,IAAI,WAAS,Y  
AAT,EAAS,oBAAT,OAAJ,C;UACI,OAAO,O;;MAEf,OAAO,I;K;8EAGX,gC;MASW,sB;;QA4FS,Q;QAAA,2B;Q  
AAhB,OAAGB,cAAhB,C;UAAgB,yB;UAAM,IA5FH,SA4FO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,  
uB;;;QAC9C,qBAAO,I;;MA7FP,yB;K;uFAGJ,gC;MAMoB,Q;MADhB,WAAe,I;MACC,2B;MAAhB,OAAGB,c  
AAhB,C;QAAGB,yB;QACZ,IA3Nc,SA2NV,CAAU,OAAV,CAAJ,C;UACI,OAAO,O;;MA5Nf,OA+NO,I;K;IA5N  
X,6B;MAQI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAc,C;QACI,MAAM,2BAAuB,oBAAvB,C;MACV,OAAO,Q  
AAS,O;K;IFAGpB,yB;MAAA,iE;MAAA,uC;QAOoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAA  
M,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,sDAAvB,C;O;KARV,C;kGAWA,  
yB;MAAA,iE;MAAA,uC;QAWW,Q;QAAA,+B;;UAcS,U;UAAA,6B;UAAhB,OAAGB,gBAAhB,C;YAAgB,2B;Y  
ACZ,aAfwB,SAeX,CAAU,OAAV,C;YACb,IAAI,cAAJ,C;cACI,8BAAO,M;cAAP,gC;;UAGR,8BAAO,I;;QApB  
A,kC;QAAA,iB;UAAmC,MAAM,gCAAuB,iEAAvB,C;;QAAhD,OAAO,I;O;KAXX,C;8GAcA,gC;MAWOB,Q;M  
AAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,aAAa,UAAU,OAAV,C;QACb,IAAI,cAAJ,C;UACI,OA  
AO,M;;MAGf,OAAO,I;K;IAGX,mC;MAMI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAc,C;QACI,OAAO,I;MAC  
X,OAAO,QAAS,O;K;6FAGpB,gC;MAMoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,  
UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;IAGX,wC;MAOiB,Q;MADb,YAAy,C;MACC,2  
B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,mBAAmB,KAAmB,C;QACA,IAAI,gBAAW,IAAX,CAAJ,C;UACI,OA  
AO,K;QACX,qB;;MAEJ,OAAO,E;K;+FAGX,yB;MAAA,wE;MAAA,uC;QAOiB,Q;QADb,YAAy,C;QACC,2B;Q  
AAb,OAAa,cAAb,C;UAAa,sB;UACT,mBAAmB,KAAmB,C;UACA,IAAI,UAAU,IAAV,CAAJ,C;YACI,OAAO,K;  
UACX,qB;;QAEJ,OAAO,E;O;KAbX,C;6FAGpB,yB;MAAA,wE;MAAA,uC;QAQiB,Q;QAFb,gBAAgB,E;QChB  
,YAAy,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,mBAAmB,KAAmB,C;UACA,IAAI,UAAU,IAAV,C  
AAJ,C;YACI,YAAy,K;UChB,qB;;QAEJ,OAAO,S;O;KAdX,C;IAiBA,4B;MAUI,eAAe,oB;MACf,IAAI,CAAC,  
QAAS,UAAc,C;QACI,MAAM,2BAAuB,oBAAvB,C;MACV,WAAW,QAAS,O;MACpB,OAAO,QAAS,UAAhB,  
C;QACI,OAAO,QAAS,O;MACpB,OAAO,I;K;+EAGX,yB;MAAA,iE;MAAA,gB;MAAA,8B;MAAA,uC;QAYoB,  
UAQT,M;QAVP,WAAe,I;QACf,YAAy,K;QACI,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,  
OAAV,CAAJ,C;YACI,OAAO,O;YACP,QAAQ,I;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,sDA  
AvB,C;QAEIB,OAAO,2E;O;KApBX,C;IAuBA,4C;MAQiB,Q;MAFb,gBAAgB,E;MACHB,YAAy,C;MACC,2B;M  
AAb,OAAa,cAAb,C;QAAa,sB;QACT,mBAAmB,KAAmB,C;QACA,IAAI,gBAAW,IAAX,CAAJ,C;UACI,YAAy,  
K;QChB,qB;;MAEJ,OAAO,S;K;IAGX,kC;MAQI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAc,C;QACI,OAAO,I;  
MACX,WAAW,QAAS,O;MACpB,OAAO,QAAS,UAAhB,C;QACI,OAAO,QAAS,O;MACpB,OAAO,I;K;2FAGX

,gC;MASoB,Q;MADhB,WAAe,I;MACC,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,OAAO,O;;;MAGf,OAAO,I,K;IAGX,8B;MAMI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QACI,MAAM,2BAAuB,oBAAvB,C;MACV,aAAa,QAAS,O;MACtB,IAAI,QAAS,UAAb,C;QACI,MAAM,gCAAyB,qCAAzB,C;MACV,OAAO,M;K;mFAGX,yB;MAAA,kF;MAAA,iE;MAAA,gB;MAAA,8B;MAAA,uC;QAQoB,UAST,M;QAXP,aAAiB,I;QACjB,YAAY,K;QACI,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,mDAazB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,sDAAvB,C;QAEIB,OAAO,6E;O;KAjBX,C;IAoBA,oC;MAMI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QACI,OAAO,I;MACX,aAAa,QAAS,O;MACtB,IAAI,QAAS,UAAb,C;QACI,OAAO,I;MACX,OAAO,M;K;+FAGX,gC;MAQoB,Q;MAFhB,aAAiB,I;MACjB,YAAY,K;MACI,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;IAGX,8B;MAWW,Q;MhBjXP,IAAI,EgBgXI,KAAK,ChBhXT,CAAJ,C;QACI,cgB+Wc,sD;QhB9Wd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MgBgXN,UAAK,CAAL,C;QAAU,gB;WACV,+C;QAAiC,OAAL,SAAK,cAAK,CAAL,C;;QACzB,wBAAa,SAAb,EAAMb,CAAnB,C;MAHZ,W;K;IAOJ,2C;MAQI,OAAO,sBAAkB,SAAlB,EAAwB,SAAxB,C;K;IAGX,wC;MAQI,OAAO,sBAAkB,SAAlB,EAAwB,IAAxB,EAA8B,SA9B,C;K;IACqE,iD;MAAA,qB;QAAE,yBAAU,EAAG,MAAb,EAAoB,EAAG,MAAvB,C;O;K;IAAkC,oC;MAAE,OAAA,EAAG,M;K;IAXzH,+C;MAWI,OAAO,yBAAqB,sBAAkB,qBAAiB,SAAjB,CAAIB,EAA0C,IAA1C,EAAGD,+BAAhD,CAArB,EAAYG,sBAAzG,C;K;oGAGX,yB;MA80BA,wE;MA90BA,oD;QAu1BiB,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UA50BT,IAAI,UA40BkB,oBAAmB,cAAAnB,EAAMb,sBAAnB,UA50BIB,EA40B+C,IA50B/C,CAAJ,C;YAA2C,sBA40BQ,IA50BR,C;;QAE/C,OAAO,W;O;KAbX,C;sGAgBA,yB;MAAA,8C;MAAA,0C;MAAA,8B;MAskB,qD;QAAA,qB;UAAE,c;S;O;MATpB,sC;QASW,Q;QAAP,OAAO,uCAAo,iCAAP,gC;O;KATX,C;0GAYA,4C;MAQoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,YAAJ,C;UAAkB,WAAy,WAAL,OAAJ,C;;MACpD,OAAO,W;K;IAGX,2C;MAQI,OAAO,sBAAkB,SAAlB,EAAwB,KAAxB,EAA+B,SAAB,C;K;IAYU,kC;MAAE,iB;K;IATvB,oC;MASW,Q;MAAP,OAAO,4CAAU,oBAAV,kC;K;IAGX,mD;MAQoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,eAAJ,C;UAAqB,WAAy,WAAL,OAAJ,C;;MACvD,OAAO,W;K;4FAGX,6C;MAQoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAy,WAAL,OAAJ,C;;MAC3D,OAAO,W;K;sFAGX,6C;MAQoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAy,WAAL,OAAJ,C;;MAC1D,OAAO,W;K;IAGX,8B;MAWW,Q;MhB1gBP,IAAI,EgBygBI,KAAK,ChBzgBT,CAAJ,C;QACI,cgBwgBc,sD;QhBvgBd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MgBygBN,UAAK,CAAL,C;QAAU,sB;WACV,+C;QAAiC,OAAL,SAAK,cAAK,CAAL,C;;QACzB,wBAAa,SAAb,EAAMb,CAAnB,C;MAHZ,W;K;IAOJ,2C;MAQI,OAAO,sBAAkB,SAAlB,EAAwB,SAAxB,C;K;IAWA,2C;MAAA,8B;K;8CACH,Y;MACI,iBAA6B,iBAAZ,gBAAy,C;MACIB,QAAX,UAAW,C;MACX,OAAO,UAAW,W;K;;IAZ9B,6B;MAQI,0C;K;sFASJ,yB;MAAA,sD;MdlfA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;Mc2ef,sC;QAUI,OAAO,sBdrfP,eAAW,iBcqfiB,QdrfjB,CAAX,CcqfO,C;O;KAVX,C;0GAaA,yB;MAAA,sD;Md5eA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,C A/EN,CAA3B,C;W;S;OA+EI,C;Mcqef,sC;QAQI,OAAO,sBd7eP,eAAW,2Bc6e2B,Qd7e3B,CAAX,Cc6eO,C;O;KARX,C;IAWA,uC;MAQI,OAAO,wBAAW,cAAX,C;K;IAWA,uE;MAAA,sC;MAAA,4C;K;kDACH,Y;MACI,iBAAiC,iBAAhB,oBAAgB,C;MACtB,WAAX,UAAW,EAAS,uBAAT,C;MACX,OAAO,UAAW,W;K;;IAZ9B,6C;MAQI,0D;K;wFASJ,yB;MAAA,wE;MAAA,uC;QAaW,kBAAy,oB;QAI FH,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAlFvB,CAAU,OAAV,C;UzBlEnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;QyBhBA,OAoFO,W;O;KAjGX,C;6FAGBA,yB;MAAA,wE;MAAA,yC;QAaW,kBAAc,oB;QA8BL,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAy,aA/B4B,WA+BxB,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QA/BhB,OAIcO,W;O;KA9CX,C;6FAGBA,yB;MAAA,wE;MAAA,yD;QAYW,kBAAc,oB;QAIcL,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAy,aAlC4B,WAKcxB,CAAY,OAAZ,CAAJ,EAlCyC,cAkCf,C AAe,OAAf,CAA1B,C;;QAIChB,OAoCO,W;O;KAhDX,C;iGAeA,+C;MAYoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,WAAy,aAAI,YAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;MAEhB,OAAO,W;K;iGAGX,

+D;MAYoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,WAAy,aAAI,YAAy,OAaZ,CAAJ,EA  
A0B,eAAe,OAaf,CAA1B,C;;MAEhB,OAAO,W;K;4FAGX,6C;MAWoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,  
C;QAAGB,yB;QACZ,WAAe,UAAU,OAAV,C;QzBIEnB,wBAAL,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;MyBoE  
A,OAAO,W;K;gGAGX,yB;MAAA,wE;MAAA,2C;QAcI,aAAa,oB;QAGBG,Q;QAAA,2B;QAAhB,OAAGB,cAAh  
B,C;UAGB,yB;UafO,MAgBP,aAAI,OAAJ,EhBe,aAgBF,CAAc,OAAd,CAAb,C;;QAhBhB,OAAuB,M;O;Kaf3  
B,C;oGakBA,iD;MAYoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,WAAy,aAAI,OAAJ,EA  
Aa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;IAGX,gD;MAMiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,s  
B;QACT,WAAy,WAAL,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gC;MAMI,OAAO,0BAAa,cAAb,C;K;IAGX,8B;M  
AMI,OAA4B,qBAAhB,iBAAL,SAAK,CAAGB,C;K;IAGhC,qC;MAMI,OAAO,0BAAa,gBAAb,C;K;IAGX,4B;MA  
QI,OAAwC,oBAAjC,0BAAa,sBAAb,CAAiC,C;K;IAG5C,0C;MAYI,OAAO,uBAAmB,SAAnB,EAAYB,SAAZB,6  
BAAoC,qB;;OApC,E;K;IAGX,0C;MAQI,OAAO,uBAAmB,SAAnB,EAAYB,SAAZB,6BAAoC,qB;;OApC,E;K;  
IAGX,iD;MAaI,OAAO,kBAaE,SAaf,EAaqB,SAArB,6BAAGC,qB;;OAAhC,E;K;IAGX,iD;MAaI,OAAO,kBAaE,  
SAaf,EAaqB,SAArB,6BAAGC,qB;;OAAhC,E;K;SAGX,yB;MAAA,wE;MAAA,gD;MAAA,oD;QAaoB,UAC4B  
,M;QAF5C,YAAy,C;QACI,2B;QAAhB,OAAGB,cAAhB,C;UAGB,yB;UACZ,WAAW,UAAU,oBAAmB,cAAAnB  
,EAAMb,sBAAnB,UAAV,EAaUC,OAAvC,C;UACC,OAAZ,WAAy,EAaO,IAAP,C;;QAEhB,OAAO,W;O;KAjB  
X,C;uGAoBA,yB;MAAA,wE;MAAA,gD;MAAA,oD;QAaoB,UAC4B,M;QAF5C,YAAy,C;QACI,2B;QAAhB,OA  
AGB,cAAhB,C;UAGB,yB;UACZ,WAAW,UAAU,oBAAmB,cAAAnB,EAAMb,sBAAnB,UAAV,EAaUC,OAAvC,  
C;UACC,OAAZ,WAAy,EAaO,IAAP,C;;QAEhB,OAAO,W;O;KAjBX,C;yFAoBA,yB;MAAA,gD;MAAA,oD;QA  
UoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAGB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAA  
Y,EAaO,IAAP,C;;QAEhB,OAAO,W;O;KAdX,C;yFAiBA,yB;MAAA,gD;MAAA,oD;QAMoB,Q;QAAA,2B;QAA  
hB,OAAGB,cAAhB,C;UAGB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAy,EAaO,IAAP,C;;QAE  
hB,OAAO,W;O;KAVX,C;qFAaA,yB;MAAA,wE;MA6BA,+D;MA7BA,yC;QAWW,kBAAU,oB;QA6BD,Q;QAA  
A,2B;QAAhB,OAAGB,cAAhB,C;UAGB,yB;UACZ,UA9BiD,WA8BvC,CAAY,OAAZ,C;UzBhoBP,U;UADP,Yy  
BkoBe,WzBloBH,WyBkoBwB,GzBloBxB,C;UACL,IAAI,aAAJ,C;YACH,ayBgoBuC,gB;YAA5B,WzB/nBX,ayB  
+nBgC,GzB/nBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UyB4nBA,iB;UACA,IAAK,WAAL,OAAJ,C;;QAhCT,OA  
kCO,W;O;KA7CX,C;qFAcA,yB;MAAA,wE;MAkCA,+D;MAiCA,yD;QAYW,kBAAU,oB;QAKCD,Q;QAAA,2B;  
QAAhB,OAAGB,cAAhB,C;UAGB,yB;UACZ,UAnCiD,WAmCvC,CAAY,OAAZ,C;UzBppBP,U;UADP,YyBspB  
e,WzBtpBH,WyBspBwB,GzBtpBxB,C;UACL,IAAI,aAAJ,C;YACH,ayBopBuC,gB;YAA5B,WzBnpBX,ayBmpBg  
C,GzBnpBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UyBgpBA,iB;UACA,IAAK,WArCyD,cAqCrD,CAAE,OAaf,C  
AAJ,C;;QArCT,OAuCO,W;O;KAnDX,C;yFAeA,yB;MAAA,+D;MAAA,sD;QAWoB,Q;QAAA,2B;QAAhB,OA  
gB,cAAhB,C;UAGB,yB;UACZ,UAAU,YAAy,OAAZ,C;UzBhoBP,U;UADP,YyBkoBe,WzBloBH,WyBkoBwB,  
GzBloBxB,C;UACL,IAAI,aAAJ,C;YACH,ayBgoBuC,gB;YAA5B,WzB/nBX,ayB+nBgC,GzB/nBhC,EAAS,MAA  
T,C;YACA,e;;YAEA,c;;UyB4nBA,iB;UACA,IAAK,WAAL,OAAJ,C;;QAET,OAAO,W;O;KAhBX,C;yFAMBA,yB  
;MAAA,+D;MAAA,sE;QAYoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAGB,yB;UACZ,UAAU,YAAy,OA  
Z,C;UzBppBP,U;UADP,YyBspBe,WzBtpBH,WyBspBwB,GzBtpBxB,C;UACL,IAAI,aAAJ,C;YACH,ayBopBuC,g  
B;YAA5B,WzBnpBX,ayBmpBgC,GzBnpBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UyBgpBA,iB;UACA,IAAK,W  
AAI,eAAe,OAaf,CAAJ,C;;QAET,OAAO,W;O;KAjBX,C;0FAoBA,yB;MAAA,kC;MAAA,4C;MAAA,wE;QAU  
W,sC;QAAA,8C;O;MAVX,oDAWQ,Y;QAA6C,OAAA,oBAAGB,W;O;MAXrE,iDAYQ,mB;QAAoC,gCAAY,OA  
AZ,C;O;MAZ5C,gF;MAAA,yC;QAUI,2D;O;KAVJ,C;IAGBA,sC;MASI,OAAO,yBAAqB,SAArB,EAa2B,SAA3B  
,C;K;IAGX,4C;MASI,OAAO,gCAA4B,SAA5B,EAakC,SAAIC,C;K;IAGX,mD;MASI,OAAoD,gBAA7C,gCAA4  
B,SAA5B,EAakC,SAAIC,CAA6C,C;K;4GAGxD,yB;MAuNA,wE;MAvNA,oD;QAGoiB,gB;QADb,YAAy,C;QA  
CC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAvNsB,U;UAAA,wBAuNT,oBAAmB,cAAAnB,EAAMb,sBAAnB,UAv  
NS,EAuNoB,IAvNpB,W;YAA6C,6B;;;QChF,OAAO,W;O;KAVX,C;8FAaA,yB;MAAA,wE;MAAA,oD;QAUiB,  
UACoC,M;QAFjD,YAAy,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAy,WAAL,UAAU,oBAAmB  
,cAAAnB,EAAMb,sBAAnB,UAAV,EAaUC,IAAvC,CAAJ,C;;QChB,OAAO,W;O;KAZX,C;IAeA,4C;MASI,OAA  
6C,gBAATC,yBAAqB,SAArB,EAa2B,SAA3B,CAAsC,C;K;8FAGjD,yB;MAAA,oD;QA4KoB,Q;QAAA,2B;QAA  
hB,OAAGB,cAAhB,C;UAGB,yB;UArKK,U;UAAA,wBAqKQ,OArKR,W;YAAsC,6B;;;QAC3D,OAAO,W;O;KA  
RX,C;IFAWA,6C;MAoiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAy,WAAL,UAAU,IAAV,C

AAJ,C;;MACHB,OAAO,W;K;IAGX,gC;MAOI,OAAO,qBAAiB,SAAJB,C;K;IACgB,6B;MAAE,S;K;IAX7B,+B;M  
AWI,OAAy,aAAL,SAAK,EAAW,eAAX,C;K;IAGhB,2C;MAYI,OAAO,qBAAiB,SAAJB,EAAB,QAABvB,C;K;I  
AGX,mC;MASiB,Q;MADb,UAAU,sB;MACG,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,GAAL,WAAL,IAAJ,C  
;;MACvB,OAAO,G;K;6EAGX,gC;MAQoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,  
CAAC,UAAU,OAAV,CAAL,C;UAAyB,OAAO,K;;MACtD,OAAO,I;K;IAGX,2B;MAQI,OAAO,oBAAW,U;K;6E  
AGtB,gC;MAQoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;U  
AAwB,OAAO,I;;MACrD,OAAO,K;K;IAGX,6B;MAOoB,Q;MADhB,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAh  
B,C;QAAGB,yB;QAAM,oBAAB,qBAAnB,EAAB,KAAnB,E;;MACtB,OAAO,K;K;iFAGX,yB;MAAA,wE;M  
AAA,uC;QAOb,Q;QADhB,YAAY,C;QACI,2B;QAAB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,O  
AAV,CAAJ,C;YAAwB,oBAAB,qBAAnB,EAAB,KAAnB,E;;QAC9C,OAAO,K;O;KARX,C;8EAWA,yC;MA  
YoB,Q;MADhB,kBAAB,O;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,cAAc,UAAU,WAAB,EA  
AuB,OAAvB,C;;MACpC,OAAO,W;K;4FAGX,yB;MAAA,wE;MAAA,gD;QAcOB,UAAiD,M;QAFjE,YAAY,C;Q  
ACZ,kBAAB,O;QACF,2B;QAAB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,cAAc,UAAU,oBAAB,cAAB,EA  
mB,sBAAB,UAAV,EAAC,WAAB,EAAD,OAAP,D,C;;QACpC,OAAO,W;O;KAFX,C;qFAkBA,6B;MAMoB,  
Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;K;kGAG1B,yB;MAAA,wE;MAA  
A,oC;QASiB,UAAgC,M;QAD7C,YAAY,C;QACC,2B;QAAB,OAAa,cAAb,C;UAAa,sB;UAAM,OAAO,oBAAB  
cAAB,EAAB,sBAAB,UAAP,EAAC,IAAP,C,C;;O;KATvB,C;IAYA,2B;MAAI,eAAe,oB;MACf,IAAI,CAAC,  
QAAS,UAAAd,C;QAAYB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,  
QAAS,O;QACjB,MZ5tCG,MAAO,KY4tCE,GZ5tCF,EY4tCO,CZ5tCP,C;;MY8tCd,OAAO,G;K;IAGX,2B;MAAI,e  
AAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAYB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAA  
S,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MZ9vCG,MAAO,KY8vCE,GZ9vCF,EY8vCO,CZ9vCP,C;;MYgwCd  
,OAAO,G;K;IAGX,2B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAYB,MAAM,6B;MAC/B,UAAU  
,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAJ,C;UAA  
a,MAAM,C;;MAEvB,OAAO,G;K;iFAGX,yB;MAAA,sE;MAAA,sC;QAAl,eAAe,oB;QACf,IAAI,CAAC,QAAS,U  
AAAd,C;UAAyB,MAAM,6B;QAC/B,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,O;QA  
ChC,eAAe,SAAS,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAA  
X,KAJ,C;YACI,UAAU,C;YACV,WAAB,C;;QAED,QAAT,QAAS,W;QACIB,OAAO,O;O;KA1BX,C;6FA6BA  
,+B;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAYB,OAAO,I;MACHC,cAAc,QAAS,O;MACvB,IAA  
I,CAAC,QAAS,UAAAd,C;QAAYB,OAAO,O;MACHC,eAAe,SAAS,OAAT,C;;QAEX,QAAQ,QAAS,O;QACjB,QA  
AQ,SAAS,CAAT,C;QACR,IAAI,2BAAW,CAAX,KAJ,C;UACI,UAAU,C;UACV,WAAB,C;;MAED,QAAT,Q  
AAS,W;MACIB,OAAO,O;K;iFAGX,yB;MAAA,sE;MZ30CA,iB;MY20CA,sC;QAeI,eAAe,oB;QACf,IAAI,CAAC  
,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UA  
CI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WZv1CG,MAAO,KYu1CO,QZv1CP,EYu1CiB,CZv1CjB,C;;QYy1Cd,  
OAAO,Q;O;KATBX,C;iFAyBA,yB;MAAA,sE;MZ/2CA,iB;MY+2CA,sC;QAeI,eAAe,oB;QACf,IAAI,CAAC,QAA  
S,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QA  
AQ,SAAS,QAAS,OAAIB,C;UACR,WZ33CG,MAAO,KY23CO,QZ33CP,EY23CiB,CZ33CjB,C;;QY63Cd,OAAO,  
Q;O;KATBX,C;iFAyBA,yB;MAAA,sE;MAAA,sC;QAAl,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,M  
AAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OA  
AIB,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,WAAB,C;;QAGnB,OAAO,Q;O;KATBX,C;6FAyBA,yB;MZ  
t5CA,iB;MYs5CA,sC;QAAl,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,  
QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WZh6CG,MAA  
O,KYg6CO,QZh6CP,EYg6CiB,CZh6CjB,C;;QYk6Cd,OAAO,Q;O;KApBX,C;6FAuBA,yB;MZx7CA,iB;MYw7C  
A,sC;QAAl,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAIB,  
C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WZl8CG,MAAO,KYk8CO,QZl8  
CP,EYk8CiB,CZl8CjB,C;;QYo8Cd,OAAO,Q;O;KApBX,C;6FAuBA,+B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QA  
AS,UAAAd,C;QAAYB,OAAO,I;MACHC,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QA  
AQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,2BAAW,CAAX,KAJ,C;UACI,WAAB,C;;MAGnB,OAAO,Q;K;yFA  
GX,yB;MAAA,sE;MAAA,kD;QAAl,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,e

AAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI,UA AW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAtB X,C;qGAYBA,2C;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,UA AW,SA AQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;IAGX,iC;MASI,e AAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,QAAS,O;MACnB,OAAO,QAAS, UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MZjhDG,MAAO,KYihDE,GZjhDF,EYihDO,CZjhDP,C;;;MYmhDd,OA AO,G;K;IAGX,iC;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,QAAS, O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MZ/iDG,MAAO,KY+iDE,GZ/iDF,EY+iDO, CZ/iDP,C;;;MYijDd,OAAO,G;K;IAGX,iC;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I; MACHC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAA N,KAAJ,C;UAAa,MAAM,C;;;MAEvB,OAAO,G;K;IAGX,2C;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd, C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACj B,IAAI,UAAW,SAAQ,GAAR,EAaA,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;;MAE9C,OAAO,G;K;I AGX,iD;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,QAAS,O;MACn B,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAAW,SAAQ,GAAR,EAaA,CAAb,CAAX,GA A6B,CAAjC,C;UAAoC,MAAM,C;;;MAE9C,OAAO,G;K;IAGX,2B;MAaI,eAAe,oB;MACf,IAAI,CAAC,QAAS,U AAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O; QACjB,MZ95CG,MAAO,KY85CE,GZ95CF,EY85CO,CZ95CP,C;;;MYg6Cd,OAAO,G;K;IAGX,2B;MAaI,eAAe,o B;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UA AhB,C;QACI,QAAQ,QAAS,O;QACjB,MZ8CG,MAAO,KYg8CE,GZ8CF,EYg8CO,CZ8CP,C;;;MYk8Cd,OAA O,G;K;IAGX,2B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAA S,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MA AM,C;;;MAEvB,OAAO,G;K;iFAGX,yB;MAAA,sE;MAAA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd, C;UAAyB,MAAM,6B;QAC/B,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,O;QAChC,e AAe,SAAS,OAAT,C;;;UAEX,QAAQ,QAAS,O;UACjB,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KA AJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAED,QAAT,QAAS,W;QACIB,OAAO,O;O;KA1BX,C;6FA6BA,+B;M ASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,cAAc,QAAS,O;MACvB,IAAI,CAA C,QAAS,UAAAd,C;QAAyB,OAAO,O;MACHC,eAAe,SAAS,OAAT,C;;;QAEX,QAAQ,QAAS,O;QACjB,QAAQ,SA AS,CAAT,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,UAAU,C;UACV,WAAW,C;;;MAED,QAAT,QAAS,W ;MACIB,OAAO,O;K;iFAGX,yB;MAAA,sE;MZ7gDA,iB;MY6gDA,sC;QAeI,eAAe,oB;QACf,IAAI,CAAC,QAAS, UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAA Q,SAAS,QAAS,OAAIB,C;UACR,WZzhDG,MAAO,KYyhDO,QZzhDP,EYyhDiB,CZzhDjB,C;;;QY2hDd,OAAO, Q;O;KAtBX,C;iFAYBA,yB;MAAA,sE;MZjjDA,iB;MYijDA,sC;QAeI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd, C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAA S,QAAS,OAAIB,C;UACR,WZ7jDG,MAAO,KY6jDO,QZ7jDP,EY6jDiB,CZ7jDjB,C;;;QY+jDd,OAAO,Q;O;KAtB X,C;iFAYBA,yB;MAAA,sE;MAAA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B; QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UA CR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAtBX,C;6FAyBA,yB;MZxlDA,iB; MYwlDA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS, OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WZlmDG,MAAO,KYk mDO,QZlmDP,EYkmDiB,CZlmDjB,C;;;QYomDd,OAAO,Q;O;KApBX,C;6FAuBA,yB;MZ1nDA,iB;MY0nDA,sC; QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAIB,C;QA Cf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WZpoDG,MAAO,KYooDO,QZpoDP, EYooDiB,CZpoDjB,C;;;QYsoDd,OAAO,Q;O;KApBX,C;6FAuBA,+B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS ,UAAAd,C;QAAyB,OAAO,I;MACHC,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAA Q,SAAS,QAAS,OAAIB,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;yFAG X,yB;MAAA,sE;MAAA,kD;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eA

Ae,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI, UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAtBX ,C;qGAyBA,2C;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,eAAe,SAAS,Q AAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,UAAW,SAA Q,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;IAGX,iC;MASI,eA Ae,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,UAAU,QAAS,O;MACnB,OAAO,QAAS,U AAhB,C;QACI,QAAQ,QAAS,O;QACjB,MZntDG,MAAO,KYmtDE,GZntDF,EYmtDO,CZntDP,C;;MYqtDd,OA AO,G;K;IAGX,iC;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,UAAU,QAAS, O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MZjvDG,MAAO,KYivDE,GZjvDF,EYivDO ,CZjvDP,C;;MYmvDd,OAAO,G;K;IAGX,iC;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAA O,I;MACHc,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,C AAN,KAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UA Ad,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;Q ACjB,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G ;K;IAGX,iD;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,UAAU,QAAS,O;M ACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX ,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,4B;MAQI,OAAO,CAAC,oBAAW,U;K;+EA GvB,gC;MAQoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UA AwB,OAAO,K;;MACrD,OAAO,I;K;IAUI,uC;MAAA,qB;QACP,eAAO,EAAP,C;QAAA,OACA,E;O;K;IATR,sC; MAOI,OAAO,kBAAI,qBAAJ,C;K;IAeW,8C;MAAA,iC;QACd,eAAO,KAAP,EAAC,oAAAd,C;QAAA,OACA,O;O; K;IAXR,6C;MASI,OAAO,wBAAW,4BAAX,C;K;kFAMX,yB;MAAA,4F;MAAA,uC;QAEI,eAAe,SAAK,W;QAC pB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,mCAA8B,kCAA9B,C;QAC/B,kBAAqB,QAAS,O;QAC9B,OA AO,QAAS,UAAhB,C;UACI,cAAc,UAAU,WAAV,EAAuB,QAAS,OAAhC,C;;QAEIB,OAAO,W;O;KArBX,C;gG AwBA,yB;MAAA,4F;MAAA,wE;MAAA,uC;QAoBmD,Q;QAL/C,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS, UAAAd,C;UAAyB,MAAM,mCAA8B,kCAA9B,C;QAC/B,YAAy,C;QACZ,kBAAqB,QAAS,O;QAC9B,OAAO,QA AS,UAAhB,C;UACI,cAAc,UAAU,oBAAmB,YAAnB,EAAMb,oBAAnB,QAAS,EAAuC,WAAvC,EAAoD,QAA S,OAA7D,C;;QAEIB,OAAO,W;O;KAtBX,C;4GAyBA,yB;MAAA,wE;MAAA,uC;QAoBmD,Q;QAL/C,eAAe,SA AK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QACHc,YAAy,C;QACZ,kBAAqB,QAAS,O;QAC 9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,oBAAmB,YAAnB,EAAMb,oBAAnB,QAAS,EAAuC,WAAvC, EAAoD,QAAS,OAA7D,C;;QAEIB,OAAO,W;O;KArBX,C;8FAyBA,gC;MAGBI,eAAe,SAAK,W;MACpB,IAAI,C AAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,kBAAqB,QAAS,O;MAC9B,OAAO,QAAS,UAAhB,C;QACI,cA Ac,UAAU,WAAV,EAAuB,QAAS,OAAhC,C;;MAEIB,OAAO,W;K;IAoBS,2I;MAAA,wC;MAAA,6B;MAAA,yB; MAAA,8C;MAAA,gD;MAAA,kD;MAAA,wB;MAAA,+B;MAAA,kC;K;;;sDAAA,Y;;;cACZ,gB;8BAAA,iCAA M,0BAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;;uCACKB,0B;cACF,wD;cAAhB,gB;;;cAAA,KAAgB,yBAhB,C ;gBAAA,gB;;;cAAgB,oC;cACZ,yBAAc,6BAAU,sBAAV,EAAuB,OAAvB,C;cACd,gB;8BAAA,iCAAM,sBAAN, O;kBAAA,2C;uBAAA,yB;cAAA,Q;;;cAFJ,gB;;;cAIJ,W;;;K;IAPgB,wF;MAAA,yD;uBAAA,+H;YAAA,S;i BAAA,Q;;iBAAA,uB;O;K;IAjBpB,sD;MAiBI,OAAO,SAAS,iDAAT,C;K;IA4BS,yJ;MAAA,wC;MAAA,6B;MAA A,yB;MAAA,8C;MAAA,8D;MAAA,kD;MAAA,wB;MAAA,yB;MAAA,+B;MAAA,kC;K;;;6DAAA,Y;;;kBAK mC,I;cAJ/C,gB;8BAAA,iCAAM,0BAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;;iCACY,C;uCACM,0B;cACF,+D; cAAhB,gB;;;cAAA,KAAgB,yBAhB,C;gBAAA,gB;;;cAAgB,oC;cACZ,yBAAc,6BAAU,oBAAmB,uBAAnB,EA AmB,+BAAnB,QAAS,EAAuC,sBAAvC,EAAoD,OAApD,C;cACd,gB;8BAAA,iCAAM,sBAAN,O;kBAAA,2C;u BAAA,yB;cAAA,Q;;;cAFJ,gB;;;cAIJ,W;;;K;IARgB,sG;MAAA,yD;uBAAA,6I;YAAA,S;iBAAA,Q;;iBAA A,uB;O;K;IAIbPb,6D;MAkBI,OAAO,SAAS,wDAAT,C;K;IA2BS,4H;MAAA,wC;MAAA,6B;MAAA,yB;MAAA, oD;MAAA,kD;MAAA,4B;MAAA,+B;MAAA,kC;K;;;wDAAA,Y;;;oCACG,wC;cACf,IAAI,mBAAS,UAAb,C;y CACyB,mBAAS,O;gBAC9B,gB;gCAAA,iCAAM,sBAAN,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBAFJ,gB;;;c AGI,gB;;;cAAA,KAAO,mBAAS,UAAhB,C;gBAAA,gB;;;cACI,yBAAc,6BAAU,sBAAV,EAAuB,mBAAS,OAAh C,C;cACd,gB;8BAAA,iCAAM,sBAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;;cAFJ,gB;;;cAHJ,gB;;;cAQJ,W;;; ;K;IAVgB,yE;MAAA,yD;uBAAA,gH;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAhBpB,+C;MAGBI,OAAO,SAAS





UAAhB,C;gBAAA,gB;;;gCACe,mBAAS,O;cACpB,gB;8BAAA,iCAAM,6BAAU,kBAAV,EAAMb,eAAnB,CAA  
N,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cACA,qBAAU,e;cAHd,gB;;;cAKJ,W;;;;;;K;IATwB,uE;MAAA,yD;u  
BAAA,4G;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAZ5B,6C;MAYI,OAAO,SAAS,0CAAT,C;K;IAYX,8F;MAU6D,  
yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MA  
AI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;MAGtN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YA  
AY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,  
SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACW,gBAAP,MAAO,EAAC,OAAd,EAAuB,SA  
vB,C;;UACJ,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,M  
AAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,4F;MAUwC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,  
E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAA  
oC,I;MACjN,OAAO,oBAAO,sBAAP,EAawB,SAaxB,EAAMc,MAAnC,EA2C,OAA3C,EAAoD,KAApD,EAA  
2D,SAA3D,EAASe,SAATe,CAAI,F,W;K;IAOXe,8C;MAAA,mB;QAAE,OAAA,eAAK,W;O;K;IAJ3B,kC;MAII,oC  
AAgB,8BAAhB,C;K;2FAGJ,qB;MAKI,OAAO,S;K;IAGX,+B;MASoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;  
MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAMb,qBAAnB,EAAMb,KAAAnB  
,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MASoB,Q;MAFhB,U  
AAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAM  
B,qBAAnB,EAAMb,KAAAnB,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IA  
GjD,+B;MASoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QA  
CZ,OAAO,O;QACP,oBAAMb,qBAAnB,EAAMb,KAAAnB,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAA  
vB,GAAgC,MAAM,K;K;IAGjD,+B;MASoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAg  
B,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAMb,qBAAnB,EAAMb,KAAAnB,E;;MAEJ,OAAW,UAAS,  
CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MASoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,  
C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAMb,qBAAnB,EAAMb,KAA  
nB,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MASoB,Q;MAFhB  
,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAA  
mB,qBAAnB,EAAMb,KAAAnB,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;I  
AGjD,2B;MAQoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,O;;MA  
EX,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAC  
Z,YAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;  
QAAgB,yB;QACZ,YAAO,OAAP,I;;MAEJ,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,Y;MACgB,2B;MAAhB,O  
AAgB,cAAhB,C;QAAgB,yB;QACZ,cAAO,OAAP,C;;MAEJ,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAiB,  
G;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAQoB,Q  
;MADhB,UAAkB,G;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;;MAEX,OAAO,G;K;IC  
xgGX,qC;MAMI,aAAa,qBAAiB,YAAY,cAAZ,CAAjB,C;MACb,kBAAC,KAAAd,C;MX8zBgB,Q;MAAA,OW7zBT  
,SX6zBS,W;MAAhB,OAAgB,cAAhB,C;QAAgB,2B;QAAU,oB;QW7zBK,IAAI,CAAC,SAAD,IAAY,OX6zBX,S  
W7zBW,UAAhB,C;UAAiC,YAAU,I;UAA3C,mBAAiD,K;;UAAjD,mBAA8D,I;;QX6zBvE,qB;UW7zBD,MX6zB  
qC,WAAI,SAAJ,C;;MW7zB1D,OAAqB,M;K;IAGzB,sC;MAUI,aAAa,qBAAiB,SAAJB,C;MACN,YAAP,MAAO,  
EAAU,QAAV,C;MACP,OAAO,M;K;IAGX,sC;MAUI,YAAqB,gCAAT,QAAS,EAAGC,SAAhC,C;MACrB,IAAI,  
KAAM,UAAV,C;QACI,OAAY,QAAL,SAAK,C;MACHb,IAAI,yBAAJ,C;QACgB,kBAAY,sB;QXmxBZ,Q;QAA  
A,OWnxBL,SXmxBK,W;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAAM,IAAI,CWnxBwB,qBXmxBb,OWnxBa,  
CXmxB5B,C;YAAyB,WAAY,WAAI,OA AJ,C;;QWnxBvD,OXoxBG,W;;MWnxBP,aAAa,qBAAiB,SAAJB,C;MA  
Cb,MAAO,mBAAU,KAAV,C;MACP,OAAO,M;K;IAGX,uC;MAUI,aAAa,qBAAiB,SAAJB,C;MACN,YAAP,MA  
AO,EAAU,QAAV,C;MACP,OAAO,M;K;gGAGX,yB;MAAA,8C;MAAA,qC;QAOL,OAAO,iBAAM,OAAN,C;O;  
KAPX,C;IAUA,qC;MAMI,aAAa,qBAAiB,YAAY,iBAAO,CAAP,IAAZ,CAAjB,C;MACb,MAAO,gBAAO,SAAP,  
C;MACP,MAAO,WAAI,OA AJ,C;MACP,OAAO,M;K;IAGX,sC;MAOI,aAAa,qBAAiB,YAAY,SAAK,KAAL,GA  
AY,QAAS,OAARb,IAAZ,CAAjB,C;MACb,MAAO,gBAAO,SAAP,C;MACA,SAAP,MAAO,EAAO,QAAP,C;MA  
CP,OAAO,M;K;IAGX,sC;MAMuD,UAAT,M;MAA1C,aAAa,qBAAiB,YAAY,WAAS,4BAAT,QAAS,CAAT,YA  
A4C,cAAL,WAAvC,4BAA2D,SAAK,KAAL,GAAY,CAAZ,IAAvE,CAAjB,C;MACb,MAAO,gBAAO,SAAP,C;

MACA,OAAP,MAAO,EAAO,QAAP,C;MACP,OAAO,M;K;IAGX,sC;MAOI,aAAa,qBAAiB,YAAY,SAAK,KAA  
L,GAAY,CAAZ,IAAZ,CAAjB,C;MACb,MAAO,gBAAO,SAAP,C;MACA,SAAP,MAAO,EAAO,QAAP,C;MACP  
,OAAO,M;K;8FAGX,yB;MAAA,4C;MAAA,qC;QAOI,OAAO,gBAAK,OAAL,C;O;KAPX,C;InBnIA,oD;MAMuF  
,wC;K;IANvF,8CAOI,Y;MAAuC,8B;K;IAP3C,gF;ICGA,oD;MAQuF,wC;K;IARvF,8CASI,Y;MAAuC,8B;K;IAT3  
C,gF;gGmBYA,yB;MAAA,uD;MAAA,gC;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAA  
C,qBAAI,KAAJ,CAATc,GAAuD,uBAAa,KAAb,E;O;KAPjE,C;gGAUA,yB;MAAA,+C;MAAA,mC;QAOI,OAA  
Y,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;0EAU,yB;MA6EA,6C;MAAA,oC;MAAA,gC;MA7EA,uC;QAOW,  
sB;;UA0ES,Q;UAAA,0B;UAAhB,OAAgB,cAAhB,C;YAAgB,oC;YAAM,IA1EH,SA0EO,CAAU,oBAAV,CAAJ,  
C;cAAwB,qBAAO,O;cAAP,uB;;;UAC9C,qBAAO,I;;;QA3EP,yB;O;KAPJ,C;kFAUA,yB;MAyJA,mD;MAAA,+C;  
MAAA,oC;MAZJA,uC;QAOW,qB;;UAwJO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAAAd,OAAc,cAAAd,C;YA  
Ac,uB;YACV,cAAc,qBAAK,KAAL,C;YACd,IA1Jc,SA0JV,CAAU,oBAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,s  
B;;;UAE5B,oBAAO,I;;;QA5JP,wB;O;KAPJ,C;IAUA,6B;MAMI,ICiOgD,qBAAU,CDjO1D,C;QACI,MAAM,2BA  
AuB,yBAAvB,C;MACV,OAAO,qBAAK,CAAL,C;K;4EAGX,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,iE;M  
AAA,uC;QAKoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;Y  
AAwB,OAAO,O;;QACrD,MAAM,gCAAuB,6DAAvB,C;O;KANV,C;6FASA,yB;MAAA,iE;MAYA,6C;MAAA,o  
C;MAAA,gC;MAZA,uC;QASW,Q;QAAA,+B;;UAYS,U;UAAA,4B;UAAhB,OAAgB,gBAAhB,C;YAAgB,sC;YA  
CZ,aAbwB,SAaX,CAAU,oBAAV,C;YACb,IAAI,cAAJ,C;cACI,8BAAO,M;cAAP,gC;;;UAGR,8BAAO,I;;;QAIBA  
,kC;QAAA,iB;UAAmC,MAAM,gCAAuB,sEAAvB,C;;QAAhD,OAAO,I;O;KATX,C;yGAYA,yB;MAAA,6C;MA  
AA,oC;MAAA,gC;MAAA,uC;QASoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,aAAa,UAAU,  
oBAAV,C;UACb,IAAI,cAAJ,C;YACI,OAAO,M;;;QAGf,OAAO,I;O;KafX,C;IAkBA,mC;MAII,OCiLgD,qBAAU,  
CDjLnD,GAAe,IAAf,GAAyB,qBAAK,CAAL,C;K;wFAGpC,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;Q  
AIoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAA  
O,O;;QACrD,OAAO,I;O;KALX,C;mFAQA,yB;MAAA,uD;MAAA,gC;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IA  
Ac,SAAS,wBAA3B,GAAc,qBAAI,KAAJ,CAATc,GAAuD,uBAAa,KAAb,E;O;KALjE,C;IAQA,uC;MAMI,OAA  
W,SAAS,CAAT,IAAc,SAAS,2BAA3B,GAAc,qBAAI,KAAJ,CAATc,GAAuD,I;K;0FAGjE,yB;MAAA,mD;MAA  
A,oC;MAAA,uC;QAikB,gC;QAAA,6B;QAAA,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,UAAU,iCAAK,  
KAAL,EAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;wFAYA,yB;MAAA,mD;MAAA,+C;MAA  
A,oC;MAAA,uC;QAikB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,IA  
AI,UAAU,iCAAK,KAAL,EAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;IAYA,4B;MAQI,ICqHg  
D,qBAAU,CDrH1D,C;QACI,MAAM,2BAAuB,yBAAvB,C;MACV,OAAO,qBAAK,2BAAL,C;K;0EAGX,yB;MA  
AA,mD;MAAA,+C;MAAA,oC;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CA  
Ab,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,cAAc,qBAAK,KAAL,C;UACd,IAAI,UAAU,oBAAV,CAAJ,C;YA  
AwB,OAAO,O;;QAEnc,MAAM,gCAAuB,6DAAvB,C;O;KAZV,C;IAeA,kC;MAMI,OC2FgD,qBAAU,CD3FnD,  
GAAe,IAAf,GAAyB,qBAAK,mBAAS,CAAT,IAAL,C;K;sFAGpC,yB;MAAA,mD;MAAA,+C;MAAA,oC;MAAA  
,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,cA  
Ac,qBAAK,KAAL,C;UACd,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnc,OAAO,I;O;KAVX,C;8E  
AaA,yB;MAAA,mC;MAAA,yC;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;IAWA,sC;MAOI,ICyDg  
D,qBAAU,CDzD1D,C;QACI,MAAM,2BAAuB,yBAAvB,C;MACV,OAAO,qBAAI,MAAO,iBAAQ,gBAAR,CAA  
X,C;K;0FAGX,yB;MAAA,mC;MAAA,qD;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;IAUA,4C;MA  
MI,ICoCgD,qBAAU,CDpC1D,C;QACI,OAAO,I;MACX,OAAO,qBAAI,MAAO,iBAAQ,gBAAR,CAAX,C;K;IAG  
X,8B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,MAAM,2BAAuB,yBAAvB,C;aACX,C;UAAK,  
4BAK,CAAL,C;UAAL,K;;UACQ,MAAM,gCAAyB,0CAAzB,C;;MAHIB,W;K;8EAOJ,yB;MAAA,6C;MAAA,o  
C;MAAA,kF;MAAA,gC;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UASt,M;QAXP,aAAoB,I;QACpB,YAAY,K;  
QACI,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,IAAI,UAAU,oBAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cA  
AW,MAAM,8BAAyB,wDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MA  
AM,gCAAuB,6DAAvB,C;QAEIB,OAAO,4E;O;KafX,C;IAkBA,oC;MAII,OAAW,qBAAU,CAAd,GAAiB,qBAA  
K,CAAL,CAAjB,GAA8B,I;K;0FAGzC,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QAMoB,Q;QAFhB,aA  
AoB,I;QACpB,YAAY,K;QACI,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,IAAI,UAAU,oBAAV,CAAJ,C;

YACI,IAAI,KAAJ,C;cAAW,OAAO,I;YACIB,SAAS,O;YACT,QAAQ,I;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,  
OAAO,I;QACnB,OAAO,M;O;KAdX,C;IAiBA,+B;MIBIRI,IAAI,EkBkSI,KAAK,CIBIST,CAAJ,C;QACI,ckBiSc,  
wD;QIBhSd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MkBiSV,OAAO,8BAAc,eAAF,CAAE,EAAa,gBAAb,CAAd,E  
AAoC,gBAAPc,C;K;IAGX,+B;MIBtSI,IAAI,EkB8SI,KAAK,CIB9ST,CAAJ,C;QACI,ckB6Sc,wD;QIB5Sd,MAA  
M,gCAAYB,OAAQ,WAAjC,C;;MkB6SV,OLx6E,oBKwF1D,eAAF,CAAE,EAAa,gBAAb,CLx6D,C;K;IK2FjF,  
kC;MIBITI,IAAI,EkB0TI,KAAK,CIB1TT,CAAJ,C;QACI,ckByTc,wD;QIBxTd,MAAM,gCAAYB,OAAQ,WAAjC,  
C;;MkByTV,OAAO,mBAAkB,gBAAZ,mBAAS,CAAT,IAAY,EAAc,CAAd,CAAIB,C;K;IAGX,mC;MIB9TI,IAAI  
,EkBsUL,KAAK,CIBtUT,CAAJ,C;QACI,ckBqUc,wD;QIBpUd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MkBqUV,OA  
AO,mBAAkB,gBAAZ,mBAAS,CAAT,IAAY,EAAc,CAAd,CAAIB,C;K;2FAGX,yB;MAAA,uD;MAAA,oC;MAA  
A,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV,CAAL,C;YA  
CI,OAAO,8BAAY,CAAZ,EAAe,QAAQ,CAAR,IAAf,C;QACf,OAAO,E;O;KATX,C;4FAYA,yB;MAAA,uD;MA  
AA,oC;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV,  
CAAL,C;YACI,OLpIoF,oBKoInE,CLpImE,EkOIH,E,QAAQ,CAAR,ILpIgE,C;;QKqI5F,OAAO,E;O;KATX,C;oFA  
YA,yB;MAAA,mD;MAAA,oC;MAAA,uC;QAMuB,UAAU,MAAK,EAAL,MAAK,EAAL,M;QAAK,mBAAL,SA  
AK,C;QAAL,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV,CAAL,C;Y  
ACI,OAAO,8BAAY,KAAZ,EAAmB,gBAAnB,C;QACf,OAAO,E;O;KATX,C;oFAYA,yB;MAAA,mD;MAAA,oC  
;MAAA,uC;QAMuB,UAAU,MAAK,EAAL,MAAK,EAAL,M;QAAK,mBAAL,SAAK,C;QAAL,mB;QAAA,kB;Q  
AAA,kB;QAAd,0D;UACI,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV,CAAL,C;YACI,OL/JqE,oBK+JpD,KL/JoD,  
C;;QKqK7E,OAAO,E;O;KATX,C;8EAYA,yB;MAAA,yD;MAkFA,oC;MAIFA,uC;QAMW,kBAAS,oB;QAKFM,Q  
;QAAA,uB;QAAtB,iBAAc,CAAd,wB;UACI,cAAc,qBAAI,KAAJ,C;UACd,IApF6B,SAoFzB,CAAU,oBAAV,CA  
AJ,C;YAAwB,WAAy,gBAAO,OAAP,C;;QApFxC,OAsFO,W;O;KA5FX,C;8EASA,yB;MAAA,yD;MAyEA,oC;  
MAzEA,uC;QAMW,kBAAS,oB;QAYEM,Q;QAAA,uB;QAAtB,iBAAc,CAAd,wB;UACI,cAAc,qBAAI,KAAJ,C;U  
ACd,IA3E6B,SA2EzB,CAAU,oBAAV,CAAJ,C;YAAwB,WAAy,gBAAO,OAAP,C;;QA3ExC,OA6EO,WA7EqC,  
W;O;KANhD,C;4FASA,yB;MAAA,yD;MA+sBA,6C;MAAA,oC;MARuBA,uC;QAQW,kBAAGB,oB;  
QAouBV,gB;QADb,YAAy,C;QACC,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UAAM,eAAO,cAAP,EAAO,sBAAP,S;  
UAAA,cAAGB,iB;UA7sB/B,IAvBoC,SAuBhC,CAAU,OAAV,EAAiB,OAAjB,CAAJ,C;YAA2C,2BAAO,kBAAP,  
C;;QAvB/C,OAYBO,W;O;KAjCX,C;4FAWA,yB;MAAA,yD;MAWA,gC;MA+sBA,6C;MAAA,oC;MAItBA,uC;Q  
AQW,kBAAGB,oB;QAYtBV,gB;QADb,YAAy,C;QACC,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UAAM,eAAO,cAA  
P,EAAO,sBAAP,S;UAAA,cAAGB,iB;UA7sB/B,IAZoC,SAYhC,CAAU,OAAV,EAAiB,OAAjB,CAAJ,C;YAA2C,  
2BAAO,kBAAP,C;;QAZ/C,OAcO,WAd4C,W;O;KARvD,C;gGAWA,yB;MAAA,gC;MA+sBA,6C;MAAA,oC;MA  
/sBA,oD;QAstBiB,gB;QADb,YAAy,C;QACC,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UAAM,eAAO,cAAP,EAAO,s  
BAAP,S;UAAA,cAAGB,iB;UA7sB/B,IAAI,UAAU,OAAV,EAAiB,OAAjB,CAAJ,C;YAA2C,2BAAO,kBAAP,C;;  
QAE/C,OAAO,W;O;KAXX,C;oFAcA,yB;MAAA,yD;MAkBA,6C;MAAA,oC;MAAA,gC;MAIBA,uC;QAMW,kB  
AAY,oB;QAKBH,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,CAIBU,SAkBT,CAAU,oBA  
AV,CAAL,C;YAAyB,WAAy,gBAAO,OAAP,C;;QAIB3D,OAmBO,W;O;KAZBX,C;oFASA,yB;MAAA,yD;MAS  
A,6C;MAAA,oC;MAAA,gC;MATA,uC;QAMW,kBAAY,oB;QASH,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UA  
AgB,oC;UAAM,IAAI,CATU,SAST,CAAU,oBAAV,CAAL,C;YAAyB,WAAy,gBAAO,OAAP,C;;QAT3D,OAuO  
,WAVwC,W;O;KANnD,C;wFASA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oD;QAMoB,Q;QAAA,0B;QAA  
hB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,CAAC,UAAU,oBAAV,CAAL,C;YAAyB,WAAy,gBAAO,OA  
P,C;;QAC3D,OAAO,W;O;KAPX,C;kFAUA,yB;MAAA,oC;MAAA,oD;QAM0B,Q;QAAA,uB;QAAtB,iBAAc,CA  
Ad,wB;UACI,cAAc,qBAAI,KAAJ,C;UACd,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,WAAy,gBAAO,OAAP,C;;Q  
AExC,OAAO,W;O;KAVX,C;IAaA,sC;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,E;MAC9B,OAAO,yBAAY,O  
AAZ,C;K;IAGX,sC;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,E;MAC9B,OAAO,uBAAU,OAAV,C;K;IAGX,s  
C;MAOc,Q;MAHV,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,  
E;MACTb,aAAa,mBAAc,IAAd,C;MACH,yB;MAAV,OAAU,cAAV,C;QAAU,mB;QACN,MAAO,gBAAO,qBAAI  
,CAAJ,CAAP,C;;MAEX,OAAO,M;K;4EAGX,yB;MAAA,8B;MAAA,uC;MAAA,qC;QAKY,Q;QAAR,OAA8B,M  
AAtB,2DAAsB,EAAM,OAAN,CAAe,W;O;KALjD,C;IAQA,+B;MIB9fI,IAAI,EkBsgBI,KAAK,CIBtgBT,CAAJ,C;  
QACI,ckBqgBc,wD;QIBpgBd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MkBqgBV,OAAO,8BAAY,CAAZ,EAAiB,eA

AF,CAAEE,EAaA,gBAAb,CAAjB,C;K;IAGX,+B;MIB1gBI,IAAI,EkBkhBI,KAAK,CIBlhBT,CAAJ,C;QACI,ckBih Bc,wD;QIBhhBd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MkBiHbBV,OLzT4F,oBKyT3E,CLzT2E,EKyTtE,eAAF,CA AE,EAaA,gBAAb,CLzTsE,C;K;IK4ThG,kC;MIBthBI,IAAI,EkB8hBI,KAAK,CIB9hBT,CAAJ,C;QACI,ckB6hBc, wD;QIB5hBd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MkB6hBV,aAAa,gB;MACb,OAAO,8BAAY,SAAW,eAAF,C AAE,EAaA,MAAb,CAAX,IAAZ,EAA6C,MAA7C,C;K;IAGX,mC;MIBniBI,IAAI,EkB2iBI,KAAK,CIB3iBT,CAA J,C;QACI,ckB0iBc,wD;QIBziBd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MkB0iBV,aAAa,gB;MACb,OLtV6E,oBKs V5D,SAAW,eAAF,CAAEE,EAaA,MAAb,CAAX,ILtV4D,C;K;2FKyVjF,yB;MAAA,uD;MAAA,oC;MAAA,uC;QA MI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,iCAAK,KAAL,EA AV,CAAL,C;YACI,OAAO, 8BAAY,QAAQ,CAAR,IAAZ,EAAuB,gBAAvB,C;;;QAGf,OAAO,8BAAY,CAAZ,EAAe,gBAaf,C;O;KAXX,C;4 FAcA,yB;MAAA,uD;MAAA,oC;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAA U,iCAAK,KAAL,EA AV,CAAL,C;YACI,OL/WqE,oBK+WpD,QAAQ,CAAR,IL/WoD,C;;;QKkX7E,OAAO,S;O; KAXX,C;oFAcA,yB;MAAA,oC;MAAA,uC;QAM0B,Q;QAAA,uB;QAAtB,iBAAc,CAAd,wB;UACI,IAAI,CAAC, UAAU,iCAAI,KA AJ,EA AV,CAAL,C;YACI,OAAO,8BAAY,CAAZ,EAAe,KAaf,C;;QAEf,OAAO,8BAAY,CAA Z,EAAe,gBAaf,C;O;KAVX,C;oFAaA,yB;MAAA,oC;MAAA,uC;QAM0B,Q;QAAA,uB;QAAtB,iBAAc,CAAd,w B;UACI,IAAI,CAAC,UAAU,iCAAI,KA AJ,EA AV,CAAL,C;YACI,OLvYoF,oBKuYnE,CLvYmE,EKuYhE,KLvY gE,C;;QKYy5F,OAAO,S;O;KAVX,C;IAaA,gC;MAII,OAAO,qBAAc,SAAd,CAAoB,U;K;kFAG/B,yB;MAAA,8B; MAAA,6C;MAAA,4B;QAKY,Q;QAAR,OAA8B,SAAtB,2DAAsB,CAAW,W;O;KAL7C,C;oFAQA,yB;MAAA,0 D;MAAA,yD;MAAA,uE;MA4EA,6C;MAAA,oC;MAAA,gC;MA5EA,uC;QAWI,eAAmC,cAApB,YAAY,gBAAZ, CAAoB,EAAC,EAAd,C;QAC5B,kBAAY,mBAAoB,QAAPB,C;QAYEH,Q;QAAA,0B;QAaHb,OAAGB,cAAhB,C; UAAgB,oC;UACZ,WA1E8C,SA0E/B,CAAU,oBAAV,C;U3B7EnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAARb ,C;;Q2BGA,OA4EO,W;O;KAXFX,C;wFAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MA6BA,6C;MAAA,oC;MAA A,gC;MA7BA,yC;QAWI,eAAmC,cAApB,YAAY,gBAAZ,CAAoB,EAAC,EAAd,C;QAC5B,kBAAC,mBAAuB,QA AvB,C;QA2BL,Q;QAAA,0B;QAaHb,OAAGB,cAAhB,C;UAAgB,oC;UACZ,WAAY,aA5BuC,WA4BnC,CAAY,o BAAZ,CAAJ,EAA0B,oBAA1B,C;;QA5BhB,OA8BO,W;O;KA1CX,C;wFAeA,yB;MAAA,0D;MAAA,yD;MAAA, uE;MA8BA,6C;MAAA,oC;MAAA,gC;MA9BA,yD;QAU1,eAAmC,cAApB,YAAY,gBAAZ,CAAoB,EAAC,EAAd, C;QAC5B,kBAAC,mBAAoB,QAAPB,C;QA6BL,Q;QAAA,0B;QAaHb,OAAGB,cAAhB,C;UAAgB,oC;UACZ,WA AY,aA9BoC,WA8BhC,CAAY,oBAAZ,CAAJ,EA9BiD,cA8BvB,CAAe,oBAaf,CAA1B,C;;QA9BhB,OAAGCO,W; O;KA3CX,C;4FAcA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,sD;QAUoB,Q;QAAA,0B;QAaHb,OAAGB,cA AhB,C;UAAgB,oC;UACZ,WAAY,aAAI,YAAY,oBAAZ,CAAJ,EAA0B,oBAA1B,C;;QAEhB,OAAO,W;O;KAbX, C;4FAGBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,sE;QAUoB,Q;QAAA,0B;QAaHb,OAAGB,cAAhB,C;U AAGB,oC;UACZ,WAAY,aAAI,YAAY,oBAAZ,CAAJ,EAA0B,eAAe,oBAaf,CAA1B,C;;QAEhB,OAAO,W;O;KA bX,C;wFAGBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oD;QASoB,Q;QAAA,0B;QAaHb,OAAGB,cAAhB, C;UAAgB,oC;UACZ,WAAE,UAAU,oBAAV,C;U3B7EnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;Q2B+ EA,OAAO,W;O;KAZX,C;4FAeA,yB;MAAA,uD;MAAA,0D;MAAA,yD;MAAA,uE;MAGBA,6C;MAAA,oC;MA AA,gC;MAhBA,2C;QAYI,aAAa,mBAA6D,cAAtC,YAAMb,aAAP,gBAAO,EAAa,GAAb,CAAnB,CAAsC,EAAC, EAAd,CAA7D,C;QAcG,Q;QAAA,0B;QAaHb,OAAGB,cAAhB,C;UAAgB,oC;UAbO,MAcP,aAAI,oBAAJ,EAd,e, aAcF,CAAc,oBAAd,CAAb,C;;QAdhB,OAAuB,M;O;KAb3B,C;+FAGBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;M AAA,wD;QAUoB,Q;QAAA,0B;QAaHb,OAAGB,cAAhB,C;UAAgB,oC;UACZ,WAAY,aAAI,oBAAJ,EAAa,cAA c,oBAAd,CAAb,C;;QAEhB,OAAO,W;O;KAbX,C;IAGBA,iD;MAIiB,Q;MAAA,4B;MAAb,OAAa,cAAb,C;QAAA, iC;QACT,WAAY,WAAl,iBAAJ,C;;MAEHb,OAAO,W;K;IAGX,iC;MAII,OAAO,2BAAa,eAAc,YAAMb,eAAP,g BAAO,EAAa,GAAb,CAAnB,CAAd,CAAb,C;K;IAGX,8B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;U AAK,kB;UAAL,K;aACA,C;UAAK,cAAO,iCAAK,CAAL,EAAP,C;UAAL,K;;UACa,wBAAL,SAAK,C;UAHV,K; ;MAAP,W;K;IAOJ,qC;MAII,OAAO,2BAAa,iBAAGB,gBAAhB,CAAb,C;K;IAGX,6B;MAMIb,IAAN,I;MAAA,Q AAM,gBAAN,C;aACH,C;UAAK,iB;UAAL,K;aACA,C;UAAK,aAAM,iCAAK,CAAL,EAAN,C;UAAL,K;;UACQ ,kCAAa,qBAAoB,YAAMb,eAAP,gBAAO,EAAa,GAAb,CAAnB,CAAPB,CAAb,C;UAHL,K;;MAAP,W;K;gFAO J,yB;MAAA,+D;MA0CA,6C;MAAA,oC;MAAA,gD;MAAA,gC;MA1CA,uC;QAMW,kBAAU,gB;QA wCD,Q;QA AA,0B;QAaHb,OAAGB,cAAhB,C;UAAgB,oC;UACZ,WAZC6B,SAyCIB,CAAU,oBAAV,C;UACC,OAAZ,WAA Y,EAAO,IAAP,C;;QA1ChB,OA4CO,W;O;KAlDX,C;8FASA,yB;MAAA,+D;MAeA,6C;MAAA,oC;MAAA,gD;M

AAA,gC;MAfA,uC;QAYW,kBAAiB,gB;QAcR,gB;QADhB,YAAy,C;QACI,0B;QAAhB,OAAgB,cAAhB,C;UAA  
gB,oC;UACZ,WafoC,SAezB,EAAU,cAAV,EAAU,sBAAV,WAAmB,oBAAnB,C;UACC,OAAZ,WAAy,EAAO,I  
AAP,C;;QAhBhB,OAKBO,W;O;KA9BX,C;kGAeA,yB;MAAA,6C;MAAA,oC;MAAA,gD;MAAA,gC;MAAA,oD;  
QAWoB,UACS,M;QAFzB,YAAy,C;QACI,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,WAAW,WAAU,c  
AAV,EAAU,sBAAV,WAAmB,oBAAnB,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAFx,C;  
oFAkBA,yB;MAAA,6C;MAAA,oC;MAAA,gD;MAAA,gC;MAAA,oD;QAIoB,Q;QAAA,0B;QAAhB,OAAgB,cA  
AhB,C;UAAgB,oC;UACZ,WAAW,UAAU,oBAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;  
O;KARX,C;gFAWA,yB;MAAA,wE;MAyBA,6C;MAAA,oC;MAAA,+D;MAAA,gC;MAzBA,yC;QASW,kBAAU,  
oB;QAYBD,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,UA1BoD,WA0B1C,CAAY,oBAAZ,C;U  
3BpjBP,U;UADP,Y2BsJBe,W3BtjBH,W2BsJbWb,G3BtjBxB,C;UACL,IAAI,aAAJ,C;YACH,a2BojBuC,gB;YAA5  
B,W3BnjBX,a2BmjBgC,G3BnjBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U2BgjBA,iB;UACA,IAAK,WAAI,oBA  
AJ,C;;QA5BT,OA8BO,W;O;KAvcX,C;gFAYA,yB;MAAA,wE;MA8BA,6C;MAAA,oC;MAAA,+D;MAAA,gC;M  
A9BA,yD;QAUW,kBAAU,oB;QA8BD,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,UA/BiD,WA  
+BvC,CAAY,oBAAZ,C;U3BtkBP,U;UADP,Y2BwkBe,W3BxkxkBH,W2BwkBwB,G3BxkxkxB,C;UACL,IAAI,aAAJ  
,C;YACH,a2BskBuC,gB;YAA5B,W3BrkBX,a2BqkBgC,G3BrkBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U2BkkB  
A,iB;UACA,IAAK,WAjCyD,cAiCrD,CAAE,oBAAf,CAAJ,C;;QAjCT,OAmCO,W;O;KA7CX,C;oFAaA,yB;MAA  
A,6C;MAAA,oC;MAAA,+D;MAAA,gC;MAAA,sD;QASoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC  
;UACZ,UAAU,YAAy,oBAAZ,C;U3BpjBP,U;UADP,Y2BsJBe,W3BtjBH,W2BsJbWb,G3BtjBxB,C;UACL,IAAI,a  
AAJ,C;YACH,a2BojBuC,gB;YAA5B,W3BnjBX,a2BmjBgC,G3BnjBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U2B  
gjBA,iB;UACA,IAAK,WAAI,oBAAJ,C;;QAET,OAAO,W;O;KADx,C;oFAiBA,yB;MAAA,6C;MAAA,oC;MAAA  
,+D;MAAA,gC;MAAA,sE;QAUoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,UAAU,YAAy,o  
BAAZ,C;U3BtkBP,U;UADP,Y2BwkBe,W3BxkxkBH,W2BwkBwB,G3BxkxkxB,C;UACL,IAAI,aAAJ,C;YACH,a2Bs  
kBuC,gB;YAA5B,W3BrkBX,a2BqkBgC,G3BrkBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U2BkkBA,iB;UACA,IA  
AK,WAAI,eAAe,oBAAf,CAAJ,C;;QAET,OAAO,W;O;KAFx,C;qFAkBA,yB;MAAA,6C;MAAA,oC;MAAA,kC;  
MAAA,4C;MAAA,wE;QAQW,sC;QAAA,8C;O;MARX,oDASQ,Y;QAAgD,OAAgB,SAAhB,oBAAgB,C;O;MAT  
xE,iDAUQ,mB;QAAuC,gCAAY,oBAAZ,C;O;MAV/C,gF;MAAA,yC;QAQI,2D;O;KARJ,C;wEAcA,yB;MAAA,g  
E;MAyEA,6C;MAAA,oC;MAAA,gC;MAzEA,uC;QAOW,kBAAM,eAAa,gBAAb,C;QAUeA,Q;QAAA,0B;QAAb,  
OAAa,cAAb,C;UAAa,iC;UACT,WAAy,WAxEmB,SAwEf,CAAU,iBAAV,CAAJ,C;;QAxEhB,OAYEO,W;O;KAh  
FX,C;sFAUA,yB;MAAA,gE;MA+BA,6C;MAAA,oC;MAAA,gC;MA/BA,uC;QAOW,kBAaA,eAAa,gBAAb,C;QA  
gCP,gB;QADb,YAAy,C;QACC,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UACT,WAAy,WAjC0B,SAiCtB,EAAU,cA  
AV,EAAU,sBAAV,WAAmB,iBAAnB,CAAJ,C;;QAjChB,OAKCO,W;O;KAzCX,C;mGAUA,yB;MAAA,+D;MAU  
A,gC;MAoLA,6C;MAAA,oC;MA9LA,uC;QAOW,kBAAoB,gB;QA8Ld,gB;QADb,YAAy,C;QACC,0B;QAAb,O  
AAa,cAAb,C;UAAa,iC;UApLsB,U;UAAA,cAVQ,SAUR,EAoLT,cApLS,EAoLT,sBApLS,WaOLA,iBApLA,W;Y  
AA6C,6B;;;QAVhF,OAwo,W;O;KAIBX,C;uGAUA,yB;MAAA,gC;MAoLA,6C;MAAA,oC;MApLA,oD;QA2LiB  
,gB;QADb,YAAy,C;QACC,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UApLsB,U;UAAA,yBAoLT,cApLS,EAoLT,sBA  
pLS,WaOLA,iBApLA,W;YAA6C,6B;;;QACHf,OAAO,W;O;KARX,C;0FAWA,yB;MAAA,6C;MAAA,oC;MAAA  
,gC;MAAA,oD;QAQiB,UACiB,M;QAF9B,YAAy,C;QACC,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UACT,WAAy,  
WAAI,WAAU,cAAV,EAAU,sBAAV,WAAmB,iBAAnB,CAAJ,C;;QACHB,OAAO,W;O;KAVX,C;qFAaA,yB;MA  
AA,+D;MAUA,gC;MA2IA,6C;MAAA,oC;MArJA,uC;QAOW,kBAaA,gB;QAKJJ,Q;QAAA,0B;QAAhB,OAAgB,c  
AAhB,C;UAAgB,oC;UA1IK,U;UAAA,cARe,SAQf,CA0IQ,oBA1IR,W;YAA6C,6B;;;QAR3D,OASO,W;O;KAhB  
X,C;yFAUA,yB;MAAA,gC;MA2IA,6C;MAAA,oC;MA3IA,oD;QA+IoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C  
;UAAgB,oC;UA1IK,U;UAAA,wBA0IQ,oBA1IR,W;YAA6C,6B;;;QAC3D,OAAO,W;O;KANX,C;4EASA,yB;MA  
AA,6C;MAAA,oC;MAAA,gC;MAAA,oD;QAKiB,Q;QAAA,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UACT,WAAy,  
WAAI,UAAU,iBAAV,CAAJ,C;;QACHB,OAAO,W;O;KAPX,C;IAe4B,4C;MAAA,mB;QAAE,iC;O;K;IAL9B,iC;  
MAKI,OAAO,qBAAiB,6BAAjB,C;K;wEAGX,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QAMoB,Q;QAA  
A,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,CAAC,UAAU,oBAAV,CAAL,C;YAAyB,OAAO,K;;  
QACiD,OAAO,I;O;KAPX,C;IAUA,2B;MAMI,OAAO,ECtwByC,qBAAU,CDswBnD,C;K;wEAGX,yB;MAAA,6C  
;MAAA,oC;MAAA,gC;MAAA,uC;QAMoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,U

AAU,oBAAV,CAAJ,C;YAAwB,OAAO,I;;QACrD,OAAO,K;O;KAPX,C;4EAUA,qB;MAKI,OAAO,gB;K;4EAGX ,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QAKoB,Q;QADhB,YAAY,C;QACI,0B;QAAhB,OAAgB,cAAh B,C;UAAgB,oC;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,qB;;QAC9C,OAAO,K;O;KANX,C;0EASA,yB; MAAA,6C;MAAA,oC;MAAA,gC;MAAA,gD;QAUoB,Q;QADhB,kBAaKB,O;QACF,0B;QAAhB,OAAgB,cAAh B,C;UAAgB,oC;UAAM,cAAc,UAAU,WAAV,EAAuB,oBAAvB,C;;QACpC,OAAO,W;O;KAXX,C;wFAcA,yB;M AAA,6C;MAAA,oC;MAAA,gC;MAAA,gD;QAYoB,UAA8B,M;QAF9C,YAAY,C;QACZ,kBAaKB,O;QACF,0B; QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAhB,EAAgC, oBAAhC,C;;QACpC,OAAO,W;O;KAbX,C;mFAgBA,yB;MAAA,uD;MAAA,oC;MAAA,gD;QAYoC,Q;QAHhC, YAAY,wB;QACZ,kBAaKB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,kCAAI,YAAJ,EAAI,oBAAJ ,SAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAdX,C;iGaiBA,yB;MAAA,uD;MAAA,oC;MAAA,gD;QAU, YAAY,wB;QACZ,kBAaKB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,iCAAI,KAA J,EAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAhBX,C;gFAmBA,yB;MAAA,6C;MAAA,oC;MAA A,gC;MAAA,oC;QAIoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,OAAO,oBAAP,C;;O;KAJ 1B,C;8FAOA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oC;QAOiB,UAAa,M;QAD1B,YAAY,C;QACC,0B;Q AAb,OAAa,cAAb,C;UAAa,iC;UAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,iBAAhB,C;;O;KAPvB,C;IAUA,2B; MAWiB,Q;MAFb,IC4BgD,qBAAU,CDg4B1D,C;QAAe,MAAM,6B;MACrB,UAAU,qBAAK,CAAL,C;MACG,k C;MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MA EvB,OAAO,G;K;4EAGX,yB;MAAA,sE;MAAA,uD;MAAA,oC;MAAA,sC;QAWI,ICp5BgD,qBAAU,CDo5B1D, C;UAAe,MAAM,6B;QACrB,cAAc,qBAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAj B,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,qBAA K,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW ,C;;QAGnB,OAAO,O;O;KAXBX,C;wFA2BA,yB;MAAA,uD;MAAA,oC;MAAA,sC;QAOI,IC36BgD,qBAAU,CD 26B1D,C;UAAe,OAAO,I;QACtB,cAAc,qBAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,C AAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,q BAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,W AAW,C;;QAGnB,OAAO,O;O;KApBX,C;4EAuBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;Md3pCA,iB;Mc2pCA,s C;QAEiB,Q;QAFb,ICx8BgD,qBAAU,CDw8B1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT ,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,WdpqCG,MAAO,Kco qCO,QdpqCP,EcoqCiB,CdpqCjB,C;;QcsqCd,OAAO,Q;O;KAnBX,C;4EAsBA,yB;MAAA,sE;MAAA,oC;MAAA,u D;Md5rCA,iB;Mc4rCA,sC;QAEiB,Q;QAFb,IC99BgD,qBAAU,CD89B1D,C;UAAe,MAAM,6B;QACrB,eAAe,SA AS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UA CR,WdrsCG,MAAO,KcqsCO,QdrsCP,EcqsCiB,CdrsCjB,C;;QcusCd,OAAO,Q;O;KAnBX,C;4EAsBA,yB;MAAA,s E;MAAA,oC;MAAA,uD;MAAA,sC;QAaiB,Q;QAFb,ICl/BgD,qBAAU,CDk/B1D,C;UAAe,MAAM,6B;QACrB,eA Ae,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT ,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAnBX,C;wFAsBA,yB;MAAA ,oC;MAAA,uD;Md7tCA,iB;Mc6tCA,sC;QAaiB,Q;QAFb,ICxgCgD,qBAAU,CDwgC1D,C;UAAe,OAAO,I;QACtB, eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EA AT,C;UACR,WdpuCG,MAAO,KcouCO,QdpuCP,EcouCiB,CdpuCjB,C;;QcsuCd,OAAO,Q;O;KAjBX,C;wFAoBA, yB;MAAA,oC;MAAA,uD;Md5vCA,iB;Mc4vCA,sC;QAaiB,Q;QAFb,IC5hCgD,qBAAU,CD4hC1D,C;UAAe,OAA O,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAA K,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAjBX,C;oFA oBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;MAAA,kD;QAaiB,Q;QAFb,ICpkCgD,qBAAU,CDokC1D,C;UAAe,M AAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,i CAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAaKB,CAAIB,CAAX,GAaKc,CAAiC,C;YACI,WA AW,C;;QAGnB,OAAO,Q;O;KAnBX,C;gGAsBA,yB;MAAA,oC;MAAA,uD;MAAA,kD;QAWiB,Q;QAFb,ICxlCg

D,qBAAU,CDw1C1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,C  
AAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAA  
X,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;IAoBA,iC;MAOiB,Q;MAFb,ICxmCgD,q  
BAAU,CDwmC1D,C;QAAe,OAAO,I;MACtB,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,iB;QA  
CI,QAAQ,qBAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;M  
AWiB,Q;MAFb,IC1nCgD,qBAAU,CD0nC1D,C;QAAe,MAAM,6B;MACrB,UAAU,qBAAK,CAAL,C;MACG,kC;  
MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,gBAAR,EAAa,cAAb,CAA  
X,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAOiB,Q;MAFb,ICxoCgD,qBAAU,CD  
woC1D,C;QAAe,OAAO,I;MACtB,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,q  
BAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,gBAAR,EAAa,cAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;  
;MAE9C,OAAO,G;K;IAGX,2B;MAWiB,Q;MAFb,IC1pCgD,qBAAU,CD0pC1D,C;QAAe,MAAM,6B;MACrB,U  
AAU,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QACR,IAAI,MAA  
M,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;4EAGX,yB;MAAA,sE;MAAA,uD;MAAA,oC;MAAA,sC;QA  
WI,IC9qCgD,qBAAU,CD8qC1D,C;UAAe,MAAM,6B;QACrB,cAAc,qBAAK,CAAL,C;QACd,gBAAqB,cAAL,S  
AAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,C;QACf,aAAU,CAAV,O  
AAa,SAAb,M;UACI,QAAQ,qBAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,KAA  
J,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;wFA2BA,yB;MAAA,uD;MAAA,oC;MA  
AA,sC;QAOI,ICrsCgD,qBAAU,CDqsC1D,C;UAAe,OAAO,I;QACtB,cAAc,qBAAK,CAAL,C;QACd,gBAAqB,cA  
AL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,C;QACf,aAAU,CAA  
V,OAAa,SAAb,M;UACI,QAAQ,qBAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,K  
AAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;4EAuBA,yB;MAAA,sE;MAAA,oC;  
MAAA,uD;MdjuCA,iB;MciuCA,sC;QAeiB,Q;QAFb,ICluCgD,qBAAU,CDkuC1D,C;UAAe,MAAM,6B;QACrB,e  
AAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EA  
AT,C;UACR,Wd1uCG,MAAO,Kc0uCO,Qd1uCP,Ec0uCiB,Cd1uCjB,C;;Qc4uCd,OAAO,Q;O;KAnBX,C;4EAsBA,y  
B;MAAA,sE;MAAA,oC;MAAA,uD;MdlwCA,iB;MckwCA,sC;QAeiB,Q;QAFb,ICxvCgD,qBAAU,CDwvC1D,C;  
UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAA  
Q,SAAS,iCAAK,CAAL,EAAT,C;UACR,Wd3wCG,MAAO,Kc2wCO,Qd3wCP,Ec2wCiB,Cd3wCjB,C;;Qc6wCd,O  
AAO,Q;O;KAnBX,C;4EAsBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;MAAA,sC;QAaiB,Q;QAFb,IC5wCgD,qBA  
AU,CD4wC1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAA  
V,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QA  
GnB,OAAO,Q;O;KAnBX,C;wFAsBA,yB;MAAA,oC;MAAA,uD;MdnyCA,iB;McmYCA,sC;QAaiB,Q;QAFb,ICly  
CgD,qBAAU,CDkyC1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU  
,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,Wd1yCG,MAAO,Kc0yCO,Qd1yCP,Ec0yCiB,Cd  
1yCjB,C;;Qc4yCd,OAAO,Q;O;KAjBX,C;wFAoBA,yB;MAAA,oC;MAAA,uD;Mdl0CA,iB;Mck0CA,sC;QAaiB,Q;  
QAFb,ICtzCgD,qBAAU,CDszC1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;Q  
AAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,Wdz0CG,MAAO,Kcy0CO,Qdz0CP,E  
cy0CiB,Cdz0CjB,C;;Qc20Cd,OAAO,Q;O;KAjBX,C;wFAoBA,yB;MAAA,oC;MAAA,uD;MAAA,sC;QAWiB,Q;Q  
AFb,ICx0CgD,qBAAU,CDw0C1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;Q  
AAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI  
,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;oFAoBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;MAAA,kD;QAaiB,Q;  
QAFb,IC91CgD,qBAAU,CD81C1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+  
B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EA  
akB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;gGAsBA,yB;MAAA,oC  
;MAAA,uD;MAAA,kD;QAWiB,Q;QAFb,ICl3CgD,qBAAU,CDk3C1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iC  
AAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IA  
AI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAj  
BX,C;IAoBA,iC;MAOiB,Q;MAFb,ICl4CgD,qBAAU,CDk4C1D,C;QAAe,OAAO,I;MACtB,UAAU,qBAAK,CAA  
L,C;MACG,kC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,M



AAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAWiB,Q;MAFb,ICp5CgD,qBAAU,CD05C1D,C;QAAe,MAAM,6B;M  
ACrB,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QACR,IAAI  
,UAAW,SAAQ,gBAAR,EAAa,cAAb,CAAX,GAA6B,CAAJ,C,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,i  
D;MAOiB,Q;MAFb,ICl6CgD,qBAAU,CDk6C1D,C;QAAe,OAAO,I;MACTb,UAAU,qBAAK,CAAL,C;MACG,kC  
;MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,gBAAR,EAAa,cAAb,CAA  
X,GAA6B,CAAJ,C,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,4B;MAMI,OCj7CgD,qBAAU,C;K;0ED07C  
9D,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QAMoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,o  
C;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,K;QACrD,OAAO,I;O;KAPX,C;8EAU,yB;MAAA,6C  
;MAAA,oC;MAAA,gC;MAAA,oC;QAKmC,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,OAAO  
,oBAAP,C;;QAARc,gB;O;KALJ,C;4FAQA,yB;MAAA,6B;MAAA,sC;MAZlBA,6C;MAAA,oC;MAAA,gC;MAylB  
A,2BAQiB,yB;QAJmBjB,6C;QAAA,oC;QAAA,gC;eAimBiB,0B;UAAA,4B;YAAE,aAAe,c;YA1lBjB,gB;YADb,  
YAAy,C;YACC,0B;YAAb,OAAa,cAAb,C;cAAa,iC;cAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,iBAAhB,C;;Y  
A0lBmB,W;W;S;OAAzB,C;MARjB,oC;QA1lBiB,gB;QADb,YAAy,C;QACC,0B;QAAb,OAAa,cAAb,C;UAAa,iC;  
UAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,iBAAhB,C;;QA0lBnB,gB;O;KARJ,C;8EAWA,yB;MAAA,4F;MA  
AA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,IC99CgD,qBAAU,CD89C1D,C;UACI,MAAM,mC  
AA8B,uCAA9B,C;QACV,kBAAkB,qBAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,  
wBAAV,EAAuB,iCAAK,KAAL,EAAvB,E;;QAEiB,OAAO,W;O;KAnBX,C;4FAsBA,yB;MAAA,4F;MAAA,uD;  
MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,ICp/CgD,qBAAU,CD0/C1D,C;UACI,MAAM,mCAA8B,u  
CAA9B,C;QACV,kBAAkB,qBAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,KA  
AV,EAAiB,wBAAjB,EAA8B,iCAAK,KAAL,EAA9B,E;;QAEiB,OAAO,W;O;KAnBX,C;wGAsBA,yB;MAAA,uD;M  
AAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,IC1gDgD,qBAAU,CD0gD1D,C;UACI,OAAO,I;QACX,kBA  
AkB,qBAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,KA  
AV,EAAiB,wBAAjB,EAA  
8B,iCAAK,KAAL,EAA9B,E;;QAEiB,OAAO,W;O;KAnBX,C;0FAsBA,yB;MAAA,uD;MAAA,oC;MAAA,gC;MA  
AA,uC;QAIbqB,Q;QAHjB,ICjiDgD,qBAAU,CDiiD1D,C;UACI,OAAO,I;QACX,kBAAkB,qBAAK,CAAL,C;QA  
CD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,wBAAV,EAAuB,iCAAK,KAAL,EAAvB,E;;QAEiB,OAA  
O,W;O;KApBX,C;uFAuBA,yB;MAAA,uD;MAAA,4F;MAAA,oC;MAAA,gC;MAAA,uC;QAE0B,UAEU,M;QAJh  
C,YAAy,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,uCAA9B,C;QACrB,kBAAkB,sBAAI,YAAJ  
,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,oBAAU,kCAAI,cAAJ,EAAI,sBAAJ,WAAV,EAA  
wB,wBAAxB,E;;QAEiB,OAAO,W;O;KAnBX,C;qGAsBA,yB;MAAA,uD;MAAA,4F;MAAA,oC;MAAA,gC;MA  
AA,uC;QAE0B,Q;QAFtB,YAAy,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,uCAA9B,C;QACrB  
,kBAAkB,sBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,oBAAU,KA  
AV,EAAiB,i  
CAAI,KA AJ,EAAjB,EAA6B,wBAA7B,E;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C;iHAuBA,yB;MAAA,uD;MA  
AA,oC;MAAA,gC;MAAA,uC;QAE0B,Q;QAFtB,YAAy,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACt  
B,kBAAkB,sBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,oBAAU,KA  
AV,EAAiB,  
iCAAI,KA AJ,EAAjB,EAA6B,wBAA7B,E;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C;mGAuBA,yB;MAAA,uD;M  
AAA,oC;MAAA,gC;MAAA,uC;QAgB0B,UAEU,M;QAJhC,YAAy,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OA  
AO,I;QACtB,kBAAkB,sBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,oBAAU,kC  
AAI,cAAJ,EAAI,sBAAJ,WAAV,EAAwB,wBAAxB,E;;QAEiB,OAAO,W;O;KApBX,C;wFAuBA,yB;MAAA,gD;  
MAAA,gE;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,gD;QAgBoB,Q;QAHhB,IClpDgD,qBAAU,CDkpD1D,C;U  
AAe,OAAO,OAAO,OAAP,C;QACgB,kBAAzB,eAAa,mBAAS,CAAT,IAAb,C;QAAiC,8B;QAA9C,afv2DO,W;Q  
ew2DP,kBAAkB,O;QACF,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,cAAc,UAAU,WAAV,EAAuB,oBA  
AvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KApBX,C;sGAuBA,yB;MAAA,gD;MAAA,gE;MAA  
A,mD;MAAA,oC;MAAA,gD;QAIbKb,gC;QAHd,IC1qDgD,qBAAU,CD0qD1D,C;UAAe,OAAO,OAAO,OAAP,C  
;QACgB,kBAAzB,eAAa,mBAAS,CAAT,IAAb,C;QAAiC,8B;QAA9C,af/3DO,W;Qeg4DP,kBAAkB,O;QACJ,6B;  
QAAA,mB;QAAA,kB;QAAA,kB;QAAAd,0D;UACI,cAAc,UAAU,KA  
AV,EAAiB,WAAjB,EAA8B,iCAAK,KAAL,  
EAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;4FAwBA,yB;MAAA,qD;MAAA,gE;M  
AAA,oC;MAAA,gC;MAAA,uC;QAgB0B,Q;QAHtB,ICjsDgD,qBAAU,CDIsD1D,C;UAAe,OAAO,W;QACtB,sBA  
AkB,qBAAK,CAAL,CAAlB,C;QACqC,kBAAxB,eAAgB,gBAAhB,C;QAAgC,sBAAI,0BAAJ,C;QAA7C,afv5DO,

W;Qew5De,uB;QAAAtB,iBAAC,CAAd,wB;UACI,gBAAC,oBAAU,0BAAV,EAAuB,iCAAK,KAAL,EAAvB,E;UA Cd,MAAO,WAAI,0BAAJ,C;;QAEX,OAAO,M;O;KApBX,C;0GAuBA,yB;MAAA,qD;MAAA,gE;MAAA,oC;MA AA,gC;MAAA,uC;QAIb0B,Q;QAHTB,ICztDgD,qBAAU,CDytD1D,C;UAAe,OAAO,W;QACtB,sBAaKB,qBAaK ,CAAL,CAAlB,C;QACqC,kBAAXB,eAAgB,gBAAhB,C;QAAgC,sBAAl,0BAAJ,C;QAA7C,af/6DO,W;Qeg7De,u B;QAAAtB,iBAAC,CAAd,wB;UACI,gBAAC,oBAAU,KAAV,EAAiB,0BAAjB,EAA8B,iCAAK,KAAL,EAA9B,E;U ACd,MAAO,WAAI,0BAAJ,C;;QAEX,OAAO,M;O;KArBX,C;0EAwBA,yB;MA9FA,gD;MAAA,gE;MAAA,6C;M AAA,oC;MAAA,gC;MA8FA,gD;QAcW,sB;;UA5FS,Q;UAHhB,IClpDgD,qBAAU,CDkpD1D,C;YAAe,qBAAO,O A+FH,OA/FG,C;YAAP,uB;;UACuB,kBAAZB,eAAa,mBAAS,CAAT,IAAb,C;UAAiC,sBA8F3B,OA9F2B,C;UAA 9C,afv2DO,W;Uew2DP,kBA6FmB,O;UA5FH,0B;UAAhB,OAAgB,cAAhB,C;YAAgB,oC;YACZ,cA2FwB,SA3F V,CAAU,WAAV,EAAuB,oBAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAwFP,yB;O;KAdJ, C;wFAiBA,yB;MAxFA,gD;MAAA,gE;MAAA,mD;MAAA,oC;MAwFA,gD;QAcW,6B;;UAtFO,gC;UAHd,IC1qD gD,qBAAU,CD0qD1D,C;YAAe,4BAAO,OAYFI,OAZFJ,C;YAAP,8B;;UACuB,kBAAZB,eAAa,mBAAS,CAAT,IA Ab,C;UAAiC,sBAwFpB,OAxFoB,C;UAA9C,af/3DO,W;Ueg4DP,kBAuF0B,O;UAtFZ,6B;UAAA,mB;UAAA,kB; UAAA,kB;UAAAd,0D;YACI,cAqF+B,SArFjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,iCAAK,KAAL,EAA9B,C; YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAKfP,gC;O;KafJ,C;4EAKBA,yB;MAAA,6C;MAAA,oC;M AAA,gC;MAAA,sC;QAOoB,Q;QADhB,UAAe,C;QACC,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,YAA O,SAAS,oBAAT,CAAP,I;;QAEJ,OAAO,G;O;KAVX,C;wFAaA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,sC; QAOoB,Q;QADhB,UAAkB,G;QACF,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,OAAO,SAAS,oBAAT,C ;QAEX,OAAO,G;O;KAVX,C;4EAaA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,sC;QAUoB,Q;QADhB,UAA oB,C;QACJ,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,OAAO,SAAS,oBAAT,C;;QAEX,OAAO,G;O;KAb X,C;4EAgBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,sC;QAUoB,Q;QADhB,UAAe,C;QACC,0B;QAAhB, OAAgB,cAAhB,C;UAAgB,oC;UACZ,YAAO,SAAS,oBAAT,CAAP,I;;QAEJ,OAAO,G;O;KAbX,C;4EAgBA,yB; MAAA,SASoB,gB;MATpB,6C;MAAA,oC;MAAA,gC;MAAA,sC;QAUoB,Q;QADhB,Y;QACgB,0B;QAAhB,OA AgB,cAAhB,C;UAAgB,oC;UACZ,cAAO,SAAS,oBAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;4EAgBA,yB;MA AA,6C;MAAA,oC;MAAA,gC;M9B/uDA,6B;M8B+uDA,sC;QAWoB,Q;QADhB,U9B/uDmC,c8B+uDnB,C9B/uD mB,C;Q8BgvDnB,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,M9BnjEiD,c8BmjEjD,G9BnjE2D,KAAK,G 8BmjEzD,SAAS,oBAAT,C9BnjEoE,KAAX,IAAf,C;;Q8BqjErD,OAAO,G;O;KAdX,C;4EAIbA,yB;MAAA,6C;M AAA,oC;MAAA,gC;Mb7vDA,+B;Ma6vDA,sC;QAWoB,Q;QADhB,Ub5vDqC,eAAW,oBA4vD/B,Cb5vD+B,CAA X,C;Qa6vDrB,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,MbjkEmD,eaiKEnD,GbjkE8D,KAAK,KaikE5D, SAAS,oBAAT,CbjkEuE,KAAX,CAAhB,C;;QamkEvD,OAAO,G;O;KAdX,C;IAiBA,oC;MAWI,OAAO,sBAAS,IA AT,EAAe,IAAf,EAAc,IAAtC,C;K;IAGX,+C;MAGBI,OAAO,sBAAS,IAAT,EAAe,IAAf,EAAc,IAAtC,EAAwD ,SAAxD,C;K;IAcsB,oC;MAAE,OAAA,EAAG,W;K;IAXtC,0C;MAWI,OAAO,6BAAgB,IAAhB,EAAc,sBAAtB, C;K;IAGX,uD;MAGBI,OAAO,8BAAiB,IAAjB,EAAuB,IAAvB,EAA8C,IAA9C,EAAgE,SAAhE,C;K;oFAGX,yB; MAAA,yD;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,oB;QACZ,aAAA, oB;QACG,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,IAAI,UAAU,oBAAV,CAAJ,C;YACI,KAAM,gBAA O,OAAP,C;;YAEN,MAAO,gBAAO,OAAP,C;;QAGf,OAAO,cAAK,KAAL,EAAy,MAAZ,C;O;KAjBX,C;oFAoB A,yB;MAAA,yD;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,oB;QACZ, aAAA,oB;QACG,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,IAAI,UAAU,oBAAV,CAAJ,C;YACI,KAAM, gBAAO,OAAP,C;;YAEN,MAAO,gBAAO,OAAP,C;;QAGf,OAAO,cAAK,KAAM,WAAx,EAAuB,MAAO,WAA 9B,C;O;KAjBX,C;IAqCgD,6B;MAAE,OAAA,EAAG,W;K;IAjBrD,2D;MAGB4C,oB;QAAA,OAAY,C;MAAG,8B ;QAAA,iBAA0B,K;MACjF,OAAO,sBAAS,IAAT,EAAe,IAAf,EAAqB,cAArB,EAAqC,eAArC,C;K;IAGX,sE;MA kBgD,oB;QAAA,OAAY,C;MAAG,8B;QAAA,iBAA0B,K;MAQhE,Q;MAPrB,oBAAoB,IAApB,EAA0B,IAA1B,C ;MACA,eAAe,SAAK,O;MACpB,qBAAqB,YAAW,IAAX,SAASB,YAAW,IAAX,UAAmB,CAAvB,GAA0B,CAA 1B,GAAiC,CAAnD,K;MACrB,aAAa,iBAAa,cAAb,C;MACb,YAAY,C;MACZ,OAAgB,CAAT,qBAAiB,QAAxB, C;QACI,UAAU,QAAQ,IAAR,I;QACO,IAAI,MAAM,CAAN,IAAW,MAAM,QAAR,C;UAAiC,IAAI,cAAJ,C;YA AoB,e;;YAAc,K;;UAAa,U;QAAjG,qB;QACA,MAAO,WAAI,UAAU,8BAAy,KAAZ,EAAmB,UAAAnB,CAAV,C AAJ,C;QACP,gBAAS,IAAT,I;;MAEJ,OAAO,M;K;IAoB6C,qC;MAAE,OAAA,EAAG,W;K;IAjB7D,iE;MAGBoD, oB;QAAA,OAAY,C;MAAG,8B;QAAA,iBAA0B,K;MACzF,OAAO,8BAAiB,IAAjB,EAAuB,IAAvB,EAA6B,cA

A7B,EAA6C,uBAA7C,C;K;IAwByB,2F;MAAA,wB;QAC5B,UAAU,QAAQ,YAAR,I;QACV,iBAAqB,MAAM,C  
AAN,IAAW,MAAM,4BAARb,GAA6B,4BAA7B,GAAyC,G;QAD1D,OAEA,kBAAU,0CAAy,KAAZ,EAAmB,U  
AAnB,CAAV,C;O;K;IAxBR,gF;MAkBWd,sB;QAAA,SAAY,C;MAAG,8B;QAAA,iBAA0B,K;MAC7F,oBAAoB,  
IAApB,EAA0B,MAA1B,C;MACA,cAAc,KAAK,cAAJ,GAAoB,yBAAPb,GAAiC,WAAQ,mBAAS,IAAT,GAAg  
B,CAAhB,IAAR,CAAIC,EAAkE,MAAIE,C;MACd,OAA4B,OAAb,aAAR,OAAQ,CAAa,EAAI,qDAAJ,C;K;IAOh  
C,kC;MAkBI,adtnEO,MAAO,KcsnEU,gBdtnEV,Ec2mEH,KAW2B,OdtExB,C;McnEd,WAAW,iBAAa,MAAb,  
C;MACX,aAAU,CAAV,MAAkB,MAAIB,M;QACI,IAAK,WAdqB,GAcP,iCAAK,CAAL,EAdO,EAcE,YAdrB,KA  
cqB,YAAM,CAAN,EAdF,CACrB,C;:MAdT,OAgBO,I;K;wEAbX,yB;MAAA,gE;MAAA,oC;MdpnEA,iB;MconEA  
,8C;QAQI,adtnEO,MAAO,KcsnEK,SAAK,OdtExB,EcsnEkB,KAAM,OdtExB,C;QcunEd,WAAW,eAAa,MAAb,  
C;QACX,aAAU,CAAV,MAAkB,MAAIB,M;UACI,IAAK,WAAI,UAAU,iCAAK,CAAL,EAAV,EAAmB,6BAAM  
,CAAN,EAAAnB,CAAJ,C;:QAET,OAAO,I;O;KAbX,C;IAgBA,kC;MASW,sB;:QAaP,WAAW,mBAAS,CAAT,I;Q  
ACX,IAAI,OAAO,CAAX,C;UAAc,qBAAO,W;UAAP,uB;:QACd,aAAa,iBAAa,IAAb,C;QACb,iBAAc,CAAd,UA  
AsB,IAAtB,U;UACI,MAAO,WAjBkB,GaiBJ,iCAAK,KAAL,EajBI,EaiBS,iCAAK,QAAQ,CAAR,IAAL,EajBT,  
CAiBlB,C;:QAEX,qBAAO,M;:;MAnBP,yB;K;uFAGJ,yB;MAAA,qD;MAAA,gE;MAAA,oC;MAAA,uC;QAUI,W  
AAW,mBAAS,CAAT,I;QACX,IAAI,OAAO,CAAX,C;UAAc,OAAO,W;QACrB,aAAa,eAAa,IAAb,C;QACb,iBA  
Ac,CAAd,UAsB,IAAtB,U;UACI,MAAO,WAAI,UAAU,iCAAK,KAAL,EAAV,EAAuB,iCAAK,QAAQ,CAAR,I  
AAL,EAAvB,CAAJ,C;:QAEX,OAAO,M;O;KAhBX,C;IAwBoB,8C;MAAA,mB;QAAE,OAAK,WAAL,eAAK,C;  
O;K;IAL3B,kC;MAIQ,wC;MAAA,S;QAAkB,OC9nE0B,qBAAU,C;:MD8nE1D,S;QAAiC,OAAO,W;MACxC,oC  
AAgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,WAAL,eAAK,C;O;K;IAL3B,kC;MAIQ,wC;MAAA,S  
;QAAkB,OCtoE0B,qBAAU,C;:MDsoE1D,S;QAAiC,OAAO,e;MACxC,oCAAgB,8BAAhB,C;K;IEh2EkC,yC;MA  
AA,wB;QAAW,OAAA,aAAK,KAAL,CjCuLV,K;O;K;IkCvLH,wC;MAAA,wB;QAAW,OAAA,aAAK,KAAL,CjC  
+NV,K;O;K;IkC/NC,yC;MAAA,wB;QAAW,OAAA,aAAK,KAAL,CjB0OV,K;O;K;IkB1OC,0C;MAAA,wB;QAA  
W,OAAA,aAAK,KAAL,CiCkMV,K;O;K;4FmC5PzC,qB;MAUI,OAAO,sBAAL,CAAJ,C;K;6FAGX,qB;MAUI,OA  
AO,sBAAL,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAL,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAL,CAAJ,C;K  
;4FAGX,qB;MAUI,OAAO,sBAAL,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAL,CAAJ,C;K;6FAGX,qB;MAUI,O  
AAO,sBAAL,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAL,CAAJ,C;K;4FAGX,qB;MAUI,OAAO,sBAAL,CAAJ,C;  
K;6FAGX,qB;MAUI,OAAO,sBAAL,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAL,CAAJ,C;K;6FAGX,qB;MAUI,  
OAAO,sBAAL,CAAJ,C;K;4FAGX,qB;MAUI,OAAO,sBAAL,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAL,CAAJ,  
C;K;6FAGX,qB;MAUI,OAAO,sBAAL,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAL,CAAJ,C;K;4FAGX,qB;MAUI  
I,OAAO,sBAAL,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAL,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAL,CAA  
J,C;K;6FAGX,qB;MAUI,OAAO,sBAAL,CAAJ,C;K;uGAuCX,yB;MAkhHI,8D;MAIhHJ,iD;QASe,oBAAS,C;QAA  
T,S;UAAc,gBAyGHT,cAAR,iBAAQ,C;:QAzGhHb,OAAO,OAAsC,sBAAL,KAAL,CAAtC,GAAsD,aAAa,KAAb,C;  
O;KATjE,C;uGAYA,yB;MA8gHI,8D;MA9gHJ,iD;QASe,oBAAS,C;QAAT,S;UAAc,gBAqgHT,cAAR,iBAAQ,C;:  
QArgHhB,OAAO,OAAsC,sBAAL,KAAL,CAAtC,GAAsD,aAAa,KAAb,C;O;KATjE,C;uGAYA,yB;MA0gHI,8D;M  
A1gHJ,iD;QASe,oBAAS,C;QAAT,S;UAAc,gBAigHT,cAAR,iBAAQ,C;:QAjgHhB,OAAO,OAAsC,sBAAL,KAAL,  
CAAtC,GAAsD,aAAa,KAAb,C;O;KATjE,C;uGAYA,yB;MAsgHI,8D;MATgHJ,iD;QASe,oBAAS,C;QAAT,S;UA  
Ac,gBA6/GT,cAAR,iBAAQ,C;:QA7/GhB,OAAO,OAAsC,sBAAL,KAAL,CAAtC,GAAsD,aAAa,KAAb,C;O;KATj  
E,C;uGAYA,yB;MAAA,sD;MAAA,mC;QASI,OAAy,UAAL,SAAK,EAAU,KAAV,C;O;KAThB,C;uGAYA,yB;  
MAAA,sD;MAAA,mC;QASI,OAAy,UAAL,SAAK,EAAU,KAAV,C;O;KAThB,C;uGAYA,yB;MAAA,sD;MAA  
A,mC;QASI,OAAy,UAAL,SAAK,EAAU,KAAV,C;O;KAThB,C;uGAYA,yB;MAAA,sD;MAAA,mC;QASI,OAA  
Y,UAAL,SAAK,EAAU,KAAV,C;O;KAThB,C;iFAYA,gC;MASW,sB;:QAkOS,Q;QAAA,2B;QAAhB,OAAgB,cA  
AhB,C;UAAgB,yB;UAAM,IAIOH,SAkOO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;:QAC9C,qBA  
AO,I;:;MAnOP,yB;K;iFAGJ,gC;MASW,sB;:QAiOS,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM  
,IAjOH,SAiOO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;:QAC9C,qBAAO,I;:;MAIOP,yB;K;iFAGJ  
,gC;MASW,sB;:QAqOS,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAhOH,SAgOO,CAAU,OA  
AV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;:QAC9C,qBAAO,I;:;MAjOP,yB;K;iFAGJ,gC;MASW,sB;:QA+NS,Q;  
QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IA/NH,SA+NO,CAAU,OAAV,CAAJ,C;YAAwB,qBA  
AO,O;YAAP,uB;:QAC9C,qBAAO,I;:;MAhOP,yB;K;yFAGJ,yB;MAgoBA,+C;MAkuFI,0D;MAI2GJ,uC;QASW,q

B;;UAgOBO,Q;UAAA,OAAa,SAytFX,YAAR,iBAAQ,CAztFW,CAAb,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,cAAc,sBAAK,KAAL,C;YACd,IAloBc,SAkoBV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAPoBP,wB;O;KATJ,C;yFAYA,yB;MAooBA,+C;MA0tFI,0D;MA91GJ,uC;QASW,qB;;UAooBO,Q;UAAA,OAAa,SAitFX,YAAR,iBAAQ,CAjtFW,CAAb,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,cAAc,sBAAK,KAAL,C;YACd,IAtoBc,SAsoBV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAxoBP,wB;O;KATJ,C;yFAYA,yB;MAwoBA,+C;MAktFI,0D;MA11GJ,uC;QASW,qB;;UAwoBO,Q;UAAA,OAAa,SAysFX,YAAR,iBAAQ,CAzsFW,CAAb,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,cAAc,sBAAK,KAAL,C;YACd,IAloBc,SA0oBV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QA5oBP,wB;O;KATJ,C;yFAYA,yB;MA4oBA,+C;MA0sFI,0D;MAt1GJ,uC;QASW,qB;;UA4oBO,Q;UAAA,OAAa,SAisFX,YAAR,iBAAQ,CAjsFW,CAAb,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,cAAc,sBAAK,KAAL,C;YACd,IA9oBc,SA8oBV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAhpBP,wB;O;KATJ,C;mFAYA,yB;MAAA,8C;MpCpHA,6B;MoCoHA,4B;QASI,OpCnHmC,coCmHpB,MAAR,iBAAQ,CpCnHoB,C;O;KoC0GvC,C;mFAYA,yB;MAAA,8C;MnBjHA,+B;MmBiHA,4B;QASI,OnBhHsC,emBgHvB,MAAR,iBAAQ,CnBhHuB,C;O;KmBuG1C,C;mFAYA,yB;MAAA,8C;MrC1LA,+B;MqC0LA,4B;QASI,OrCzLsC,eqCyLvB,MAAR,iBAAQ,CrCzLuB,C;O;KqCgL1C,C;mFAYA,yB;MAAA,8C;MnCzLA,iC;MmCyLA,4B;QASI,OnCxLyC,gBmCwL1B,MAAR,iBAAQ,CnCXL0B,C;O;KmC+K7C,C;mFAYA,yB;MAAA,iE;MAAA,uC;QAQoB,Q;QAAA,2B;QAahB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KATV,C;mFAYA,yB;MAAA,iE;MAAA,uC;QAQoB,Q;QAAA,2B;QAahB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KATV,C;mFAYA,yB;MAAA,iE;MAAA,uC;QAQoB,Q;QAAA,2B;QAahB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KATV,C;mFAYA,yB;MAAA,iE;MAAA,uC;QAQoB,Q;QAAA,2B;QAahB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KATV,C;IAYA,mC;MAMI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,CAAL,C;K;IAGpC,mC;MAMI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,CAAL,C;K;IAGpC,mC;MAMI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,CAAL,C;K;+FAGpC,gC;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;+FAGX,gC;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;+FAGX,gC;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;+FAGX,gC;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;2FAGX,yB;MAkqGI,8D;MAIqGJ,iD;QAOe,oBAAS,C;QAAT,S;UAAc,gBA2pGT,cAAR,iBAAQ,C;;QA3pGhB,OAAO,OAAAsC,sBAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KAPjE,C;2FAUA,yB;MAgqGI,8D;MAhqGJ,iD;QAOe,oBAAS,C;QAAT,S;UAAc,gBAypGT,cAAR,iBAAQ,C;;QAZpGhB,OAAO,OAAAsC,sBAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KAPjE,C;2FAUA,yB;MA8pGI,8D;MA9pGJ,iD;QAOe,oBAAS,C;QAAT,S;UAAc,gBAupGT,cAAR,iBAAQ,C;;QAVpGhB,OAAO,OAAAsC,sBAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KAPjE,C;2FAUA,yB;MA4pGI,8D;MA5pGJ,iD;QAOe,oBAAS,C;QAAT,S;UAAc,gBAqpGT,cAAR,iBAAQ,C;;QArpGhB,OAAO,OAAAsC,sBAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KAPjE,C;IAUA,wC;MAQe,oBAAS,C;MAAT,S;QAAC,gBAknGT,gBAAR,iBAAQ,C;;MAlnGhB,OAAO,OAAAsC,sBAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,wC;MAQe,oBAAS,C;MAAT,S;QAAC,gBA4mGT,gBAAR,iBAAQ,C;;MA5mGhB,OAAO,OAAAsC,sBAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,wC;MAQe,oBAAS,C;MAAT,S;QAAC,gBAymGT,gBAAR,iBAAQ,C;;MAzmGhB,OAAO,OAAAsC,sBAAI,KAAJ,CAAtC,GAAsD,I;K;uFAGjE,yB;MAAA,kD;MAAA,qC;QAOI,OAAe,QAAR,iBAAQ,EAAQ,OpC1dU,KoC0dlB,C;O;KAPnB,C;uFAUA,yB;MAAA,kD;MAAA,qC;QAOI,OAAe,QAAR,iBAAQ,EAAQ,OrCthBY,KqCshBpB,C;O;KAPnB,C;uFAUA,yB;MAAA,kD;MAAA,qC;QAOI,OAAe,QAAR,iBAAQ,EAAQ,OnCrhBc,KmCqhBtB,C;O;KAPnB,C;uFAUA,yB;MAAA,sC;MpChaA,6B;MoCgaA,0BAOGC,yB;QpCvahC,6B;eoCuagC,6B;UAAA,qB;YAAE,yBpC7ZK,coC6ZK,EpC7ZL,CoC6ZL,C;W;S;OAAF,C;MAPhC,uC;QAOmB,kBAAR,iB;QAA

Q,uB;;UvC+0Bf,0D;YACl,IuCh1B0B,UpC7ZK,cH6uCjB,YAAK,KAAL,CG7uCiB,CoC6ZL,CvCg1B1B,C;cACl,sBAAO,K;cAAP,wB;;;UAGR,sBAAO,E;;;QuCp1BP,0B;O;KAPJ,C;iGAUA,yB;MAAA,sC;MnB3ZA,+B;MmB2ZA,0BAOgC,yB;QnBlahC,+B;emBkagC,6B;UAAA,qB;YAAE,yBnBxZQ,emBwZE,EnBxZF,CmBwZR,C;W;S;OAAF,C;MAPhC,uC;QAOMb,kBAAR,iB;QAAQ,uB;;UvCi1Bf,0D;YACl,IuCl1B0B,UnBxZQ,epB0uCpB,YAAK,KAAL,CoB1uCoB,CmBwZR,CvCk1B1B,C;cACl,sBAAO,K;cAAP,wB;;;UAGR,sBAAO,E;;;QuCt1BP,0B;O;KAPJ,C;iGAUA,yB;MAAA,sC;MrCleA,+B;MqCkeA,0BAOgC,yB;QrCzehC,+B;eqCyegC,6B;UAAA,qB;YAAE,yBrC/dQ,eqC+dE,ErC/dF,CqC+dR,C;W;S;OAAF,C;MAPhC,uC;QAOMb,kBAAR,iB;QAAQ,uB;;UvCmyBf,0D;YACl,IuCpyB0B,UrC/dQ,eFmwCpB,YAAK,KAAL,CEnwCoB,CqC+dR,CvCoyB1B,C;cACl,sBAAO,K;cAAP,wB;;;UAGR,sBAAO,E;;;QuCxyBP,0B;O;KAPJ,C;iGAUA,yB;MAAA,sC;MnC/dA,iC;MmC+dA,0BAOgC,yB;QnCtehC,iC;emCsegC,6B;UAAA,qB;YAAE,yBnC5dW,gBmC4dD,EnC5dC,CmC4dX,C;W;S;OAAF,C;MAPhC,uC;QAOMb,kBAAR,iB;QAAQ,uB;;UvCqyBf,0D;YACl,IuCtyB0B,UnC5dW,gBJkwCvB,YAAK,KAAL,CllwCuB,CmC4dX,CvCsyB1B,C;cACl,sBAAO,K;cAAP,wB;;;UAGR,sBAAO,E;;;QuC1yBP,0B;O;KAPJ,C;+FAUA,yB;MAAA,sC;MvCs5BA,0D;MAAA,+C;MG91CA,6B;MoCwcA,yBAO+B,yB;QpC/c/B,6B;eoC+c+B,6B;UAAA,qB;YAAE,yBpCrcM,coCqcI,EpCrcJ,CoCqcN,C;W;S;OAAF,C;MAP/B,uC;QAOMb,kBAAR,iB;QAAQ,sB;;UvCm5BD,Q;UAAA,OAAQ,SAAR,wBAAQ,CAAR,W;UAAAd,OAAC,cAAAd,C;YAAc,uB;YACV,IuCp5ByB,UpCrcM,cHy1CjB,YAAK,KAAL,CGz1CiB,CoCqcN,CvCo5BzB,C;cACl,qBAAO,K;cAAP,uB;;;UAGR,qBAAO,E;;;QuCx5BP,yB;O;KAPJ,C;+FAUA,yB;MAAA,sC;MvCw5BA,0D;MAAA,+C;MoB31CA,+B;MmBmcA,yBAO+B,yB;QnB1c/B,+B;emB0c+B,6B;UAAA,qB;YAAE,yBnBhcS,emBgcC,EnBhcD,CmBgcT,C;W;S;OAAF,C;MAP/B,uC;QAOMb,kBAAR,iB;QAAQ,sB;;UvCq5BD,Q;UAAA,OAAQ,SAAR,wBAAQ,CAAR,W;UAAAd,OAAC,cAAAd,C;YAAc,uB;YACV,IuCt5ByB,UnBhcS,epBs1CpB,YAAK,KAAL,CoBt1CoB,CmBgcT,CvCs5BzB,C;cACl,qBAAO,K;cAAP,uB;;;UAGR,qBAAO,E;;;QuC15BP,yB;O;KAPJ,C;+FAUA,yB;MAAA,sC;MvC02BA,0D;MAAA,+C;MEp3CA,+B;MqC0gBA,yBAO+B,yB;QrCjhB/B,+B;eqCihB+B,6B;UAAA,qB;YAAE,yBrCvgBS,eqCugBC,ErCvgBD,CqCugBT,C;W;S;OAAF,C;MAP/B,uC;QAOMb,kBAAR,iB;QAAQ,sB;;UvCu2BD,Q;UAAA,OAAQ,SAAR,wBAAQ,CAAR,W;UAAAd,OAAC,cAAAd,C;YAAc,uB;YACV,IuCx2ByB,UrCvgBS,eF+2CpB,YAAK,KAAL,CE/2CoB,CqCugBT,CvCw2BzB,C;cACl,qBAAO,K;cAAP,uB;;;UAGR,qBAAO,E;;;QuC52BP,yB;O;KAPJ,C;+FAUA,yB;MAAA,sC;MvC42BA,0D;MAAA,+C;MIn3CA,iC;MmCugBA,yBAO+B,yB;QnC9gB/B,iC;emC8gB+B,6B;UAAA,qB;YAAE,yBnCpgBY,gBmCogBF,EnCpgBE,CmCogBZ,C;W;S;OAAF,C;MAP/B,uC;QAOMb,kBAAR,iB;QAAQ,sB;;UvCy2BD,Q;UAAA,OAAQ,SAAR,wBAAQ,CAAR,W;UAAAd,OAAC,cAAAd,C;YAAc,uB;YACV,IuC12ByB,UnCpgBY,gBJ82CvB,YAAK,KAAL,Ci92CuB,CmCogBZ,CvC02BzB,C;cACl,qBAAO,K;cAAP,uB;;;UAGR,qBAAO,E;;;QuC92BP,yB;O;KAPJ,C;iFAUA,yB;MAAA,4C;MpChfA,6B;MoCgfa,4B;QAWI,OpCjfmC,coCifpB,KAAR,iBAAQ,CpCjfoB,C;O;KoCsevC,C;iFACa,yB;MAAA,4C;MnB/eA,+B;MmB+eA,4B;QAWI,OnBhfsC,emBgfvB,KAAR,iBAAQ,CnBhfuB,C;O;KmBqe1C,C;iFACa,yB;MAAA,4C;MrC1jBA,+B;MqC0jBA,4B;QAWI,OrC3jBsC,eqC2jBvB,KAAR,iBAAQ,CrC3jBuB,C;O;KqCgjB1C,C;iFACa,yB;MAAA,4C;MnC3jBA,iC;MmC2jBA,4B;QAWI,OnC5jByC,gBmC4jB1B,KAAR,iBAAQ,CnC5jB0B,C;O;KmCijB7C,C;iFACa,yB;MAAA,+C;MAAA,iE;MA83FI,0D;MA93FJ,uC;QAWkB,Q;QAAA,OAAa,SAm3FX,YAn3FF,SAm3FN,QAAQ,CAn3FW,CAAb,W;QAAAd,OAAC,cAAAd,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEEnC,MAAM,gCAAuB,mDAAvB,C;O;KafV,C;iFakBA,yB;MAAA,+C;MAAA,iE;MAo3FI,0D;MAp3FJ,uC;QAWkB,Q;QAAA,OAAa,SAy2FX,YAz2FF,SAy2FN,QAAQ,CAz2FW,CAAb,W;QAAAd,OAAC,cAAAd,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEEnC,MAAM,gCAAuB,mDAAvB,C;O;KafV,C;iFakBA,yB;MAAA,+C;MAAA,iE;MAg2FI,0D;MAh2FJ,uC;QAWkB,Q;QAAA,OAAa,SAq1FX,YAr1FF,SAq1FN,QAAQ,CAr1FW,CAAb,W;QAAAd,OAAC,cAAAd,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEEnC,MAAM,gCAAuB,mDAAvB,C;O;KafV,C;+FakBA,yB;MAAA,0D;MAAA,qC;QAOI,OAAe,YAAR,iBAAQ,EAAy,OpCltBM,KoCktBIB,C;O;KAPnB,C;+FAUA,yB;MAAA,0D;MAAA,qC;QAOI,OAAe,YAAR,iBAAQ,EAAy,OrC9wBQ,KqC8wBpB,C;O;KAPnB,C;+FAUA,yB;MAAA,0D;MAAA,qC;QAOI,OAAe,YAAR,iBAAQ,EAAy,OnC7w



C,oC;MAMI,OAAW,mBAAQ,CAAZ,GAAe,sBAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,oC;MAMI,OAAW,mBA  
AQ,CAAZ,GAAe,sBAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,oC;MAMI,OAAW,mBAAQ,CAAZ,GAAe,sBAAK,C  
AAL,CAAf,GAA4B,I;K;iGAGvC,gC;MASoB,Q;MAFhB,aAAoB,I;MACpB,YAA Y,K;MACl,2B;MAAhB,OAAg  
B,cAAhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACl,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,  
O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MASoB,  
Q;MAFhB,aAAqB,I;MACrB,YAA Y,K;MACl,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,UAAU,O  
AAV,CAAJ,C;UACl,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,K  
AAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MASoB,Q;MAFhB,aAAqB,I;MACrB,YAA Y,K;MACl,  
2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACl,IAAI,KAAJ,C;YAAW,O  
AAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;i  
GAGX,gC;MASoB,Q;MAFhB,aAA sB,I;MACtB,YAA Y,K;MACl,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QA  
CZ,IAAI,UAAU,OAAV,CAAJ,C;UACl,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAG  
hB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;IAGX,+B;MxBzhDI,IAAI,EwBmiDI,KAAK,Cx  
BniDT,CAAJ,C;QACl,cwBkiDc,sD;QxBjiDd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MwBkiDV,OAAO,uBAAoB,g  
BAAV,iBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,+B;MxBviDI,IAAI,EwBijDI,KAAK,CxBjjiDT,CA  
AJ,C;QACl,cwBgiDc,sD;QxB/iDd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MwBgiDV,OAAO,uBAAoB,gBAAV,iB  
AAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,+B;MxBriDI,IAAI,EwB+jDI,KAAK,CxB/jDT,CAAJ,C;QA  
Cl,cwB8jDc,sD;QxB7jDd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MwB8jDV,OAAO,uBAAoB,gBAAV,iBAAO,CA  
AP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,+B;MxBnkDI,IAAI,EwB6kDI,KAAK,CxB7kDT,CAAJ,C;QACl,cwB  
4kDc,sD;QxB3kDd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MwB4kDV,OAAO,uBAAoB,gBAAV,iBAAO,CAAP,IA  
AU,EAAC,CAAd,CAApB,C;K;IAGX,mC;MxBjldI,IAAI,EwB2lDI,KAAK,CxB3lDT,CAAJ,C;QACl,cwB0lDc,sD;  
QxBzldD,MAAM,gCAAYB,OAAQ,WAAjC,C;;MwB0lDV,OAAO,mBAAgB,gBAAV,iBAAO,CAAP,IAAU,EAAC,  
CAAd,CAAhB,C;K;IAGX,mC;MxB/ldI,IAAI,EwBymDI,KAAK,CxBzmDT,CAAJ,C;QACl,cwBwmDc,sD;QxB  
vmDd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MwBwmDV,OAAO,mBAAgB,gBAAV,iBAAO,CAAP,IAAU,EAAC,  
CAAd,CAAhB,C;K;IAGX,mC;MxB7mDI,IAAI,EwBunDI,KAAK,CxBvnDT,CAAJ,C;QACl,cwBsnDc,sD;QxBrn  
Dd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MwBsnDV,OAAO,mBAAgB,gBAAV,iBAAO,CAAP,IAAU,EAAC,CAA  
d,CAAhB,C;K;IAGX,mC;MxB3nDI,IAAI,EwBqoDI,KAAK,CxBroDT,CAAJ,C;QACl,cwBooDc,sD;QxBnoDd,M  
AAM,gCAAYB,OAAQ,WAAjC,C;;MwBooDV,OAAO,mBAAgB,gBAAV,iBAAO,CAAP,IAAU,EAAC,CAAd,CA  
AhB,C;K;mGAGX,yB;MAAA,4C;MAAA,qD;MAkqEI,8D;MAIqEJ,uC;QASI,iBAypEgB,cAAR,iBAAQ,CAzpEh  
B,WAA+B,CAA/B,U;UACl,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACl,OAAO,gBAAK,QAAQ,  
CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAdX,C;mGAiBA,yB;MAAA,4C;MAAA,qD;MAypEI,8D;MAzpEJ,uC;QA  
SI,iBAgpEgB,cAAR,iBAAQ,CAhpEhB,WAA+B,CAA/B,U;UACl,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CA  
AL,C;YACl,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAdX,C;mGAiBA,yB;MAAA,4C;MA  
AA,qD;MAgpEI,8D;MAhpEJ,uC;QASI,iBAuoEgB,cAAR,iBAAQ,CAvoEhB,WAA+B,CAA/B,U;UACl,IAAI,CA  
AC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACl,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;  
KAdX,C;mGAiBA,yB;MAAA,4C;MAAA,qD;MAuoEI,8D;MAvoEJ,uC;QASI,iBA8nEgB,cAAR,iBAAQ,CA9nEh  
B,WAA+B,CAA/B,U;UACl,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACl,OAAO,gBAAK,QAAQ,  
CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAdX,C;2FAiBA,yB;MAAA,+D;MAAA,uC;QAWiB,Q;QAFb,eAAe,K;QA  
Cf,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,QAAJ,C;YACl,IAAK,WAAI,IAAJ,C;eA  
CJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;;QAE nB,OAAO,I;O;KAI  
BX,C;2FAqBA,yB;MAAA,+D;MAAA,uC;QAWiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACE,2B;QAAb,OAAa,  
cAAb,C;UAAa,sB;UACT,IAAI,QAAJ,C;YACl,IAAK,WAAI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;  
YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;;QAE nB,OAAO,I;O;KAI BX,C;2FAqBA,yB;MAAA,+D;MAAA,u  
C;QAWiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,QAAJ,C;  
YACl,IAAK,WAAI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WA  
AW,I;;;QAE nB,OAAO,I;O;KAI BX,C;2FAqBA,yB;MAAA,+D;MAAA,uC;QAWiB,Q;QAFb,eAAe,K;QACf,WAA  
W,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,QAAJ,C;YACl,IAAK,WAAI,IAAJ,C;eACJ,IAAI,  
CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;;QAE nB,OAAO,I;O;KAI BX,C;qF





UAAZ,C;QAAuB,OIC3jEe,W;;MkC4jEtC,OAA4D,SA0iDrD,cAAkB,cAAR,iBAAQ,EA1iDN,OAAQ,MA0iDF,E  
A1iDS,OAAQ,aAAR,GAAuB,CAAvB,IA0iDT,CAAIB,CA1iDqD,C;K;IAGhE,sC;MAMI,IAAI,OAAQ,UAAZ,C;  
QAAuB,OICrKee,W;;MkCskEtC,OAA4D,SAgjDrD,eAAmB,cAAR,iBAAQ,EAhjDP,OAAQ,MAgjDD,EAhjDQ,O  
AAQ,aAAR,GAAuB,CAAvB,IAgjDR,CAAnB,CAhjDqD,C;K;IAGhE,sC;MAMI,IAAI,OAAQ,UAAZ,C;QAAuB,  
OIC/kEe,W;;MkCglEtC,OAA4D,UAsjDrD,eAAmB,cAAR,iBAAQ,EAtdjDP,OAAQ,MAtdjDD,EAtdjDQ,OAAQ,aA  
R,GAAuB,CAAvB,IASjDR,CAAnB,CAtjDqD,C;K;IAGhE,sC;MAMI,IAAI,OAAQ,UAAZ,C;QAAuB,OICZIEe,W;  
;MkC0IEtC,OAA4D,UA4jDrD,gBAAoB,cAAR,iBAAQ,EA5jDR,OAAQ,MA4jDA,EA5jDO,OAAQ,aAAR,GAAu  
B,CAAvB,IA4jDP,CAApB,CA5jDqD,C;K;IAGhE,sC;MAskB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAA  
xB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACTB,WAAW,iBAAgB,IAAhB,C;MACG,yB;MAAd,OA  
Ac,cAAd,C;QAAC,uB;QACV,IAAK,WAAI,sBAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAskB,Q;MA  
Hd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACTB,WAA  
W,iBAAiB,IAAjB,C;MACG,yB;MAAd,OAAC,cAAd,C;QAAC,uB;QACV,IAAK,WAAI,sBAAI,KAAJ,CAAJ,C;;M  
AET,OAAO,I;K;IAGX,sC;MAskB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ  
,CAAZ,C;QAAe,OAAO,W;MACTB,WAAW,iBAAiB,IAAjB,C;MACG,yB;MAAd,OAAC,cAAd,C;QAAC,uB;QAC  
V,IAAK,WAAI,sBAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAskB,Q;MAHd,WAAmB,wBAAR,OAA  
Q,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACTB,WAAW,iBAAkB,IAAiB,C;MAC  
G,yB;MAAd,OAAC,cAAd,C;QAAC,uB;QACV,IAAK,WAAI,sBAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,2C  
;MAMI,OAAO,cAAkB,aAAR,iBAAQ,EAAW,OAAx,CAAI,C;K;IAGX,2C;MAMI,OAAO,eAAmB,aAAR,iBAA  
AQ,EAAW,OAAx,CAAnB,C;K;IAGX,2C;MAMI,OAAO,eAAmB,aAAR,iBAAQ,EAAW,OAAx,CAAnB,C;K;IA  
GX,2C;MAMI,OAAO,gBAAoB,aAAR,iBAAQ,EAAW,OAAx,CAApB,C;K;IAGX,2C;MAMI,OAAO,cAAkB,cA  
AR,iBAAQ,EAAW,OAAx,CAAI,C;K;IAGX,2C;MAMI,OAAO,eAAmB,cAAR,iBAAQ,EAAW,OAAx,CAAnB,  
C;K;IAGX,2C;MAMI,OAAO,eAAmB,aAAR,iBAAQ,EAAW,OAAx,CAAnB,C;K;IAGX,2C;MAMI,OAAO,gBA  
AoB,cAAR,iBAAQ,EAAW,OAAx,CAApB,C;K;IAGX,+B;MAGBiB,Q;MxBjyEb,IAAI,EwB2xEI,KAAK,CxB3xE  
T,CAAJ,C;QACI,cwB0xEc,sD;QxBzxEd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwB0xEV,IAAI,MAAK,CAAT,C  
;QAAY,OAAO,W;MACnB,IAAI,KAAK,cAAT,C;QAAe,OAAO,mB;MACTB,IAAI,MAAK,CAAT,C;QAAY,OAA  
O,OAAO,sBAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAgB,CAAhB,C;MACE,2B;MAAb,OA  
Aa,cAAb,C;QAAa,sB;QACT,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAF,C;UACI,K;;MAER,OAAO,I;K;I  
AGX,+B;MAGBiB,Q;MxBzzEb,IAAI,EwBmzEI,KAAK,CxBnzET,CAAJ,C;QACI,cwBkzEc,sD;QxBjzEd,MAAM,  
gCAAyB,OAAQ,WAAjC,C;;MwBkzEV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,cAAT,C;  
QAAe,OAAO,mB;MACTB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,CAAL,CAAP,C;MACnB,YAA  
Y,C;MACZ,WAAW,iBAAiB,CAAjB,C;MACE,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAK,WAAI,IAAJ,C  
;QACL,IAAI,mCAAW,CAAF,C;UACI,K;;MAER,OAAO,I;K;IAGX,+B;MAGBiB,Q;MxBj1Eb,IAAI,EwB20EI,KA  
AK,CxB30ET,CAAJ,C;QACI,cwB00Ec,sD;QxBz0Ed,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwB00EV,IAAI,MA  
AK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,cAAT,C;QAAe,OAAO,mB;MACTB,IAAI,MAAK,CAAT,C;  
QAAY,OAAO,OAAO,sBAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAiB,CAAjB,C;MACE,2B;  
MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAF,C;UACI,K;;MAER,O  
AAO,I;K;IAGX,+B;MAGBiB,Q;MxBz2Eb,IAAI,EwBm2EI,KAAK,CxBn2ET,CAAJ,C;QACI,cwBk2Ec,sD;QxBj2  
Ed,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwBk2EV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KA  
AK,cAAT,C;QAAe,OAAO,mB;MACTB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,CAAL,CAAP,C;M  
ACnB,YAAY,C;MACZ,WAAW,iBAAkB,CAAI,C;MACE,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAK,W  
AAI,IAAJ,C;QACL,IAAI,mCAAW,CAAF,C;UACI,K;;MAER,OAAO,I;K;IAGX,mC;MxBj3EI,IAAI,EwB23EI,KA  
AK,CxB33ET,CAAJ,C;QACI,cwB03Ec,sD;QxBz3Ed,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwB03EV,IAAI,MA  
AK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,c;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACTB,IAAI  
,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAgB,CAAh  
B,C;MACX,iBAAC,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;MACT,  
OAAO,I;K;IAGX,mC;MxBt4EI,IAAI,EwBg5EI,KAAK,CxBh5ET,CAAJ,C;QACI,cwB+4Ec,sD;QxB94Ed,MAAM  
,gCAAyB,OAAQ,WAAjC,C;;MwB+4EV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,c;MACX,IA  
AI,KAAK,IAAT,C;QAAe,OAAO,mB;MACTB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,OAAO,CA

AP,IAAL,CAAP,C;MACnB,WAAW,iBAAiB,CAAjB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;Q  
ACI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,mC;MxB35EI,IAAI,EwBq6EI,KAAK,CxB  
6ET,CAAJ,C;QACI,cwBo6Ec,sD;QxBn6Ed,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwBo6EV,IAAI,MAAK,CAA  
T,C;QAAY,OAAO,W;MACnB,WAAW,c;MACX,IAAI,KAAK,IAAT,C;QAe,OAAO,mB;MACTB,IAAI,MAAK,  
CAAT,C;QAAY,OAAO,OAAO,sBAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAiB,CAAjB,C;MAC  
X,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;MACT,OAAO,I;K  
;IAGX,mC;MxBh7EI,IAAI,EwB07EI,KAAK,CxB17ET,CAAJ,C;QACI,cwBy7Ec,sD;QxBx7Ed,MAAM,gCAAyB,  
OAAQ,WAAjC,C;;MwBy7EV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,c;MACX,IAAI,KAAK,  
IAAT,C;QAe,OAAO,mB;MACTB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,OAAO,CAAP,IAAL,C  
AAP,C;MACnB,WAAW,iBAakB,CAAIB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,  
WAAI,sBAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;mGAGX,yB;MAAA,4C;MAAA,gD;MAS2CI,8D;MA2CJ,uC;  
QASI,iBA61CgB,cAAR,iBAAQ,CA71ChB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV  
,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAdX,C;mGaiBA,yB;MAAA,4C;  
MAAA,gD;MA61CI,8D;MA71CJ,uC;QASI,iBAo1CgB,cAAR,iBAAQ,Cap1ChB,WAA+B,CAA/B,U;UACI,IAAI,  
CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;  
O;KAdX,C;mGaiBA,yB;MAAA,4C;MAAA,gD;MAo1CI,8D;MAP1CJ,uC;QASI,iBA20CgB,cAAR,iBAAQ,CA30  
ChB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAA  
Q,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAdX,C;mGaiBA,yB;MAAA,4C;MAAA,gD;MA20CI,8D;MA30CJ,uC;Q  
ASI,iBAk0CgB,cAAR,iBAAQ,CA10ChB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,C  
AAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAdX,C;2FAiBA,yB;MAAA,+D;MA  
AA,uC;QAUiB,Q;QADb,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,CAAC,UAAU,IAA  
V,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO,I;O;KafX,C;2FAkBA,yB;MAAA,+D;MAAA,uC  
;QAUiB,Q;QADb,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,CAAC,UAAU,IAAV,CA  
AL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO,I;O;KafX,C;2FAkBA,yB;MAAA,+D;MAAA,uC;QA  
UiB,Q;QADb,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,CAAC,UAAU,IAAV,CAAL,C  
;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO,I;O;KafX,C;2FAkBA,yB;MAAA,+D;MAAA,uC;QA  
UiB,Q;QADb,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,  
K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO,I;O;KafX,C;uFAkBA,yB;MAAA,kD;MAAA,4B;QAOY,QAAR,iB  
AAQ,C;O;KAPZ,C;uFAUA,yB;MAAA,kD;MAAA,4B;QAOY,QAAR,iBAAQ,C;O;KAPZ,C;uFAUA,yB;MAAA,k  
D;MAAA,4B;QAOY,QAAR,iBAAQ,C;O;KAPZ,C;uFAUA,yB;MAAA,kD;MAAA,4B;QAOY,QAAR,iBAAQ,C;  
O;KAPZ,C;uFAUA,yB;MAAA,kD;MAAA,gD;QAaY,QAAR,iBAAQ,EAAQ,SAAR,EAAMB,OAAAnB,C;O;KAbZ,  
C;uFAGBA,yB;MAAA,kD;MAAA,gD;QAaY,QAAR,iBAAQ,EAAQ,SAAR,EAAMB,OAAAnB,C;O;KAbZ,C;uFAG  
BA,yB;MAAA,kD;MAAA,gD;QAaY,QAAR,iBAAQ,EAAQ,SAAR,EAAMB,OAAAnB,C;O;KAbZ,C;uFAGBA,yB;  
MAAA,kD;MAAA,gD;QAaY,QAAR,iBAAQ,EAAQ,SAAR,EAAMB,OAAAnB,C;O;KAbZ,C;IAGBA,gC;MAMI,IA  
AI,mBAAJ,C;QAe,OAAO,W;MACTB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,gC;MAMI  
,IAAI,mBAAJ,C;QAe,OAAO,W;MACTB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,gC;MA  
MI,IAAI,mBAAJ,C;QAe,OAAO,W;MACTB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,gC;  
MAMI,IAAI,mBAAJ,C;QAe,OAAO,W;MACTB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;kGAG  
X,yB;MAAA,8D;MAAA,uC;MAAA,4B;QAOI,OAAO,mBAakB,cAAR,iBAAQ,CAAlB,C;O;KAPX,C;kGAUA,y  
B;MAAA,8D;MAAA,yC;MAAA,4B;QAOI,OAAO,oBAAMB,cAAR,iBAAQ,CAAnB,C;O;KAPX,C;mGAUA,yB;  
MAAA,8D;MAAA,yC;MAAA,4B;QAOI,OAAO,oBAAMB,cAAR,iBAAQ,CAAnB,C;O;KAPX,C;mGAUA,yB;M  
AAA,8D;MAAA,2C;MAAA,4B;QAOI,OAAO,qBAAoB,cAAR,iBAAQ,CAApB,C;O;KAPX,C;IAUA,+B;MAMI,s  
BAAQ,4BAAR,C;K;IAGJ,+B;MAMI,sBAAQ,4BAAR,C;K;IAGJ,+B;MAMI,sBAAQ,4BAAR,C;K;IAGJ,+B;MA  
MI,sBAAQ,4BAAR,C;K;IAGJ,uC;MAQI,aA8+BgB,gBAAR,iBAAQ,CA9+BhB,OAA2B,CAA3B,M;QACI,QAAQ  
,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,sBAAK,CAAL,C;QACX,sBAAK,CAAL,EAAU,sBAAK,CA  
AL,CAAV,C;QACA,sBAAK,CAAL,EAAU,IAAV,C;;K;IAIR,uC;MAQI,aAs+BgB,gBAAR,iBAAQ,CAt+BhB,OA  
A2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,sBAAK,CAAL,C;QACX,sBA  
AK,CAAL,EAAU,sBAAK,CAAL,CAAV,C;QACA,sBAAK,CAAL,EAAU,IAAV,C;;K;IAIR,uC;MAQI,aA89BgB,

gBAAR,iBAAQ,CA99BhB,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAA  
W,sBAAK,CAAL,C;QACX,sBAAK,CAAL,EAAU,sBAAK,CAAL,CAAV,C;QACA,sBAAK,CAAL,EAAU,IAAV  
,C;;K;IAIR,uC;MAQI,aAs9BgB,gBAAR,iBAAQ,CAt9BhB,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IA  
AI,CAAJ,IAAR,C;QACf,WAAW,sBAAK,CAAL,C;QACX,sBAAK,CAAL,EAAU,sBAAK,CAAL,CAAV,C;QAC  
A,sBAAK,CAAL,EAAU,IAAV,C;;K;IAIR,sC;MAMI,IAAI,iBAAO,CAAX,C;QACI,iB;QApSI,UAAR,iBAAQ,C;;  
K;IAySZ,sC;MAMI,IAAI,iBAAO,CAAX,C;QACI,iB;QAtSI,UAAR,iBAAQ,C;;K;IA2SZ,sC;MAMI,IAAI,iBAAO,  
CAAX,C;QACI,iB;QAxSI,UAAR,iBAAQ,C;;K;IA6SZ,sC;MAMI,IAAI,iBAAO,CAAX,C;QACI,iB;QA1SI,UAAR  
,iBAAQ,C;;K;IA+SZ,6B;MAMoB,kBA+nBT,cAAU,iBvBh9EO,QuBg9EjB,C;MA/nBiB,mB;MAAxB,OAAiC,SrB  
33F1B,WqB23F0B,C;K;IAGrC,8B;MAMoB,kBAkoBT,eAAmB,UAAR,iBAAQ,CAAnB,C;MAlOBiB,mB;MAAxB  
,OAAiC,SrBp4F1B,WqB4F0B,C;K;IAGrC,8B;MAMoB,kBAqoBT,eAAW,iBvB5/EM,QuB4/EjB,C;MAroBiB,m  
B;MAAxB,OAAiC,UrB74F1B,WqB64F0B,C;K;IAGrC,8B;MAMoB,kBAwoBT,gBAAY,iBvB9/EK,QuB8/EjB,C;  
MAxoBiB,mB;MAAxB,OAAiC,UrBt5F1B,WqBs5F0B,C;K;IAGrC,kC;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,S;  
MACD,kBA0iBd,cA11BA,SA0IBU,QvBh9EO,QuBg9EjB,C;MA11BsB,mB;MAA7B,OrBh6FO,W;K;IqBm6FX,kC;  
MAMI,IAAI,mBAAJ,C;QAAe,OAAO,S;MACD,kBA4iBd,eAAmB,UA5iBnB,SA4iBW,QAAQ,CAAnB,C;MA5iB  
sB,mB;MAA7B,OrB16FO,W;K;IqB66FX,kC;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,S;MACD,kBA8iBd,eA9iBA,  
SA8iBW,QvB5/EM,QuB4/EjB,C;MA9iBsB,mB;MAA7B,OrBp7FO,W;K;IqBu7FX,mC;MAMI,IAAI,mBAAJ,C;Q  
AAe,OAAO,S;MACD,kBAgmBd,gBAhmBA,SAgmBY,QvB9/EK,QuB8/EjB,C;MAhmBsB,mB;MAA7B,OrB97F  
O,W;K;IqBi8FX,4C;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,S;MACD,kBAkjBd,cAljBA,SAkjBU,QvBh9EO,QuBg  
9EjB,C;MAIjBsB,8B;MAA7B,OrBx8FO,W;K;IqB28FX,4C;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,S;MACD,kBA  
ojBd,eAAmB,UApjBnB,SAojBW,QAAQ,CAAnB,C;MApjBsB,8B;MAA7B,OrB19FO,W;K;IqBq9FX,4C;MAMI,I  
AAI,mBAAJ,C;QAAe,OAAO,S;MACD,kBASjBd,eAtjBA,SAsjBW,QvB5/EM,QuB4/EjB,C;MATjBsB,8B;MAA7B  
,OrB59FO,W;K;IqB+9FX,6C;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,S;MACD,kBAwjBd,gBAxjBA,SAwjBY,QvB  
9/EK,QuB8/EjB,C;MAxjBsB,8B;MAA7B,OrBt+FO,W;K;IqBy+FX,uC;MAQoB,kBAygBT,cAAU,iBvBh9EO,QuB  
g9EjB,C;MAzgBiB,mB;MAAxB,OAAiC,YrBj/F1B,WqBi/F0B,C;K;IAGrC,wC;MAQoB,kBA0gBT,eAAmB,UAA  
R,iBAAQ,CAAnB,C;MA1gBiB,mB;MAAxB,OAAiC,YrB5/F1B,WqB4/F0B,C;K;IAGrC,wC;MAQoB,kBA2gBT,e  
AAW,iBvB5/EM,QuB4/EjB,C;MA3gBiB,mB;MAAxB,OAAiC,YrBvgG1B,WqBugG0B,C;K;IAGrC,wC;MAQoB,  
kBA4gBT,gBAAY,iBvB9/EK,QuB8/EjB,C;MA5gBiB,mB;MAAxB,OAAiC,YrBlhG1B,WqBkhG0B,C;K;4FAGrC,  
qB;MAQI,OAAO,iB;K;0FAGX,qB;MAQI,OAAO,iB;K;4FA+BX,qB;MAQI,OAAO,iB;K;8FAGX,qB;MAQI,OAA  
O,iB;K;8FAGX,yB;MAAA,yC;MAAA,4B;QAQI,OAAO,oBAAW,SAAX,C;O;KARX,C;4FAWA,yB;MAAA,uC;  
MAAA,4B;QAQI,OAAO,mBAAU,SAAV,C;O;KARX,C;8FAWA,yB;MAAA,yC;MAAA,4B;QAQI,OAAO,oBAA  
W,SAAX,C;O;KARX,C;gGAWA,yB;MAAA,2C;MAAA,4B;QAQI,OAAO,qBAAW,SAAZ,C;O;KARX,C;IAWA,2  
C;MASI,OAAW,gBAAL,SAAK,EAAC,KAAd,C;K;IAGhB,2C;MASI,OAAW,gBAAL,SAAK,EAAC,KAAd,C;K;IA  
GhB,2C;MASI,OAAW,gBAAL,SAAK,EAAC,KAAd,C;K;IAGhB,2C;MASI,OAAW,gBAAL,SAAK,EAAC,KAAd,  
C;K;IAGhB,2C;MAOI,OAAqB,cAAAd,4CAAc,EAAC,oCAAd,C;K;IAGzB,2C;MAOI,OAAqB,cAAAd,4CAAc,  
EAAC,oCAAd,C;K;IAGzB,2C;MAOI,OAAqB,cAAAd,4CAAc,EAAC,oCAAd,C;K;IAGzB,2C;MAOI,OAAqB,cAAAd,4CA  
Ac,EAAC,oCAAd,C;K;IAGzB,sC;MAQI,OAAW,kBAAL,SAAK,C;K;IAGhB,sC;MAQI,OAAW,kBAAL,SAAK,C;  
K;IAGhB,sC;MAQI,OAAW,kBAAL,SAAK,C;K;IAGhB,sC;MAQI,OAAW,kBAAL,SAAK,C;K;IAGhB,sC;MAMI,  
OAAqB,gBAAd,4CAAc,C;K;IAGzB,sC;MAMI,OAAqB,gBAAd,4CAAc,C;K;IAGzB,sC;MAMI,OAAqB,gBAAd,  
4CAAc,C;K;IAGzB,sC;MAMI,OAAqB,gBAAd,4CAAc,C;K;IAGzB,sC;MAUI,OAAW,kBAAL,SAAK,C;K;IAGh  
B,sC;MAUI,OAAW,kBAAL,SAAK,C;K;IAGhB,sC;MAUI,OAAW,kBAAL,SAAK,C;K;IAGhB,sC;MAUI,OAAW,k  
BAAL,SAAK,C;K;IAGhB,sC;MAQW,Q;MAAP,OAAO,sDAAmB,IAAnB,EAAYB,GAAZB,EAA8B,GAA9B,2BA  
AsC,M;K;IAGjD,sC;MAQW,Q;MAAP,OAAO,sDAAmB,IAAnB,EAAYB,GAAZB,EAA8B,GAA9B,2BAAsC,M;K;  
IAGjD,sC;MAQW,Q;MAAP,OAAO,sDAAmB,IAAnB,EAAYB,GAAZB,EAA8B,GAA9B,2BAAsC,M;K;IAGjD,s  
C;MAQW,Q;MAAP,OAAO,sDAAmB,IAAnB,EAAYB,GAAZB,EAA8B,GAA9B,2BAAsC,M;K;sFAGjD,yB;MvB  
5hFA,8C;MuB4hFA,kF;QAmB6D,iC;UAAA,oBAAyB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAg  
B,c;QvB3hF1H,UuB4hFA,iBvB5hFA,EuB4hFiB,WAAW,QvB5hF7B,EuB4hFsC,iBvB5hFiC,EuB4hFyD,UvB5hFz  
D,EuB4hFqE,QvB5hFrE,C;QuB6hFA,OAAO,W;O;KArBX,C;wFAwBA,yB;MvB5hFA,8C;MuB4hFA,kF;QAmB+  
D,iC;UAAA,oBAAyB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,c;QvB3hF5H,UuB4hFA,iBvB5

hFA, EuB4hFiB, WAAY, QvB5hF7B, EuB4hFsC, iBvB5hFtC, EuB4hFyD, UvB5hFzD, EuB4hFqE, QvB5hFrE, C; QuB6  
hFA, OAAO, W; O; KArBX, C; wFAwBA, yB; MvB5nFA, 8C; MuB4nFA, kF; QAmB+D, iC; UAAA, oBAAYB, C; QAAG,  
0B; UAAA, aAAkB, C; QAAG, wB; UAAA, WAAgB, c; QvB3nF5H, UuB4nFA, iBvB5nFA, EuB4nFiB, WAAY, QvB5nF  
7B, EuB4nFsC, iBvB5nFtC, EuB4nFyD, UvB5nFzD, EuB4nFqE, QvB5nFrE, C; QuB6nFA, OAAO, W; O; KArBX, C; wF  
AwBA, yB; MvB5nFA, 8C; MuB4nFA, kF; QAmBiE, iC; UAAA, oBAAYB, C; QAAG, 0B; UAAA, aAAkB, C; QAAG, wB;  
UAAA, WAAgB, c; QvB3nF9H, UuB4nFA, iBvB5nFA, EuB4nFiB, WAAY, QvB5nF7B, EuB4nFsC, iBvB5nFtC, EuB4n  
FyD, UvB5nFzD, EuB4nFqE, QvB5nFrE, C; QuB6nFA, OAAO, W; O; KArBX, C; kFAwBA, yB; MAAA, uC; MAAA, 4B;  
QASI, OAAO, mBAAU, iBvBh9EO, QuBg9EjB, C; O; KATX, C; oFAYA, yB; MAAA, gD; MAAA, yC; MAAA, 4B; QASI,  
OAAO, oBAAmB, OAAr, iBAAQ, CAAnB, C; O; KATX, C; oFAYA, yB; MAAA, yC; MAAA, 4B; QASI, OAAO, oBAA  
W, iBvB5/EM, QuB4/EjB, C; O; KATX, C; oFAYA, yB; MAAA, 2C; MAAA, 4B; QASI, OAAO, qBAAY, iBvB9/EK, QuB  
8/EjB, C; O; KATX, C; oFAYA, yB; MAAA, gD; MAAA, uC; MAAA, qC; QAWI, OAAO, mBAAkB, OAAr, iBAAQ, EAA  
O, OAAp, CAAlB, C; O; KAXX, C; oFACa, yB; MAAA, gD; MAAA, yC; MAAA, qC; QAWI, OAAO, oBAAmB, OAAr, iB  
AAQ, EAAO, OAAp, CAAnB, C; O; KAXX, C; oFACa, yB; MAAA, +C; MAAA, yC; MAAA, qC; QAWI, OAAO, oBAAm  
B, OAAr, iBAAQ, EAAO, OAAp, CAAnB, C; O; KAXX, C; oFACa, yB; MAAA, gD; MAAA, 2C; MAAA, qC; QAWI, OA  
AO, qBAAoB, OAAr, iBAAQ, EAAO, OAAp, CAApB, C; O; KAXX, C; 4FACa, yB; MAAA, 0D; MAAA, uC; MAAA, gD;  
QAaI, OAAO, mBAAkB, YAAR, iBAAQ, EAAY, SAAZ, EAAuB, OAAvB, CAAlB, C; O; KabX, C; 8FAGBA, yB; MAA  
A, 0D; MAAA, yC; MAAA, gD; QAaI, OAAO, oBAAmB, YAAR, iBAAQ, EAAY, SAAZ, EAAuB, OAAvB, CAAnB, C; O  
; KabX, C; 8FAGBA, yB; MAAA, 0D; MAAA, yC; MAAA, gD; QAaI, OAAO, oBAAmB, YAAR, iBAAQ, EAAY, SAAZ, E  
AAuB, OAAvB, CAAnB, C; O; KabX, C; 6FAGBA, yB; MAAA, 0D; MAAA, 2C; MAAA, gD; QAaI, OAAO, qBAAoB, YA  
AR, iBAAQ, EAAY, SAAZ, EAAuB, OAAvB, CAApB, C; O; KabX, C; IAGBA, sD; MAWyC, yB; QAAA, YAAiB, C; MA  
AG, uB; QAAA, UAAe, c; MACHe, OAAr, iBAAQ, EAAK, OpC38GoB, KoC28GzB, EAAsB, SAAtB, EAAiC, OAAjC,  
C; K; IAGZ, wD; MAW2C, yB; QAAA, YAAiB, C; MAAG, uB; QAAA, UAAe, c; MACIE, OAAr, iBAAQ, EAAK, OnB/8  
GsB, KmB+8G3B, EAAuB, SAAvB, EAAkC, OAAIC, C; K; IAGZ, wD; MAW2C, yB; QAAA, YAAiB, C; MAAG, uB; QA  
AA, UAAe, c; MACIE, OAAr, iBAAQ, EAAK, OrCjhHsB, KqCihH3B, EAAuB, SAAvB, EAAkC, OAAIC, C; K; IAGZ, w  
D; MAW6C, yB; QAAA, YAAiB, C; MAAG, uB; QAAA, UAAe, c; MACpE, OAAr, iBAAQ, EAAK, OnCrhHwB, KmCqh  
H7B, EAAwB, SAAxB, EAAmC, OAAnc, C; K; 8FASR, yB; MAAA, 0D; MAAA, 4B; QAAQ, OAAQ, YAAR, iBAAQ, C;  
O; KAAhB, C; 8FAQA, yB; MAAA, 0D; MAAA, 4B; QAAQ, OAAQ, YAAR, iBAAQ, C; O; KAAhB, C; +FAQA, yB; MAA  
A, 0D; MAAA, 4B; QAAQ, OAAQ, YAAR, iBAAQ, C; O; KAAhB, C; +FAQA, yB; MAAA, 0D; MAAA, 4B; QAAQ, OAA  
Q, YAAR, iBAAQ, C; O; KAAhB, C; kGAQA, yB; MAAA, 8D; MAAA, 4B; QAAQ, OAAQ, cAAR, iBAAQ, C; O; KAAhB,  
C; kGAQA, yB; MAAA, 8D; MAAA, 4B; QAAQ, OAAQ, cAAR, iBAAQ, C; O; KAAhB, C; mGAQA, yB; MAAA, 8D; MA  
AA, 4B; QAAQ, OAAQ, cAAR, iBAAQ, C; O; KAAhB, C; mGAQA, yB; MAAA, 8D; MAAA, 4B; QAAQ, OAAQ, cAAR, i  
BAAQ, C; O; KAAhB, C; iFAEJ, yB; MAAA, uC; MvB3oEA, iD; MuB2oEA, qC; QAOqB, 4B; QAAA, gBAAU, OpClkHM  
, K; QoCkkHjC, OAAO, mBvB7oEA, 2BAxIK, gBAAW, SAAX, EAwIL, CuB6oEA, C; O; KAPX, C; iFAUA, yB; MAAA,  
yC; MvB7oEA, iD; MuB6oEA, qC; QAOI, OAAO, oBvB/oEA, qBuB+oEW, iBvB/oEX, EAxIK, mBuBuxEgB, OnBjkHO  
, KJ0yCvB, CAwIL, CuB+oEA, C; O; KAPX, C; iFAUA, yB; MAAA, yC; MvB/qEA, iD; MuB+qEA, qC; QAOsB, 4B; QAA  
A, gBAAU, OrC9nHO, K; QqC8nHnC, OAAO, oBvBjrEA, 2BAxIK, eAAY, SAAZ, EAwIL, CuBirEA, C; O; KAPX, C; iFA  
UA, yB; MAAA, 2C; MvBjrEA, iD; MuBirEA, qC; QAOuB, 4B; QAAA, gBAAU, OnC7nHQ, K; QmC6nHrC, OAAO, qBv  
BnrEA, 2BAxIK, gBAaA, SAAb, EAwIL, CuBmrEA, C; O; KAPX, C; IAUA, sC; MAQoB, UAAiB, M; MAFjC, YAAY, c;  
MACZ, aAAqB, UAAR, iBAAQ, EAAO, iBAAO, QAAS, KAAhB, IAAP, C; MACL, 0B; MAAhB, OAAgB, cAAhB, C; Q  
AAgB, yB; QAAU, OAAO, cAAP, EAAO, sBAAP, YAAkB, OpC3mHX, K; MoC4mHjC, OAAO, cAAU, MAAV, C; K; I  
AGX, sC; MAQoB, UAAiB, M; MAFjC, YAAY, c; MACZ, aAAqB, UAAR, iBAAQ, EAAO, iBAAO, QAAS, KAAhB, IA  
AP, C; MACL, 0B; MAAhB, OAAgB, cAAhB, C; QAAgB, yB; QAAU, OAAO, cAAP, EAAO, sBAAP, YAAkB, OnB5mH  
T, K; MmB6mHnC, OAAO, eAAW, MAAX, C; K; IAGX, sC; MAQoB, UAAiB, M; MAFjC, YAAY, c; MACZ, aAAqB, U  
AAR, iBAAQ, EAAO, iBAAO, QAAS, KAAhB, IAAP, C; MACL, 0B; MAAhB, OAAgB, cAAhB, C; QAAgB, yB; QAAU,  
OAAO, cAAP, EAAO, sBAAP, YAAkB, OrC3qHT, K; MqC4qHnC, OAAO, eAAW, MAAX, C; K; IAGX, sC; MAQoB, U  
AAiB, M; MAFjC, YAAY, c; MACZ, aAAqB, UAAR, iBAAQ, EAAO, iBAAO, QAAS, KAAhB, IAAP, C; MACL, 0B; MA  
AhB, OAAgB, cAAhB, C; QAAgB, yB; QAAU, OAAO, cAAP, EAAO, sBAAP, YAAkB, OnC5qHP, K; MmC6qHrC, OA  
AO, gBAAY, MAAZ, C; K; iFAGX, yB; MAAA, uC; MvBnuEA, iD; MuBmuEA, sC; QAOI, OAAO, mBvBruEA, qBuBqu

EU,iBvBruEV,EuBquEoB,QAAS,QvBruE7B,CuBquEA,C;O;KAPX,C;iFAUA,yB;MAAA,yC;MvBruEA,iD;MuBquEA,sC;QAOI,OAAO,oBvBvuEA,qBuBuuEW,iBvBvuEX,EuBuuEqB,QAAS,QvBvuE9B,CuBuuEA,C;O;KAPX,C;iFAUA,yB;MAAA,yC;MvBvwEA,iD;MuBuwEA,sC;QAOI,OAAO,oBvBzwEA,qBuBywEW,iBvBzwEX,EuBywEqB,QAAS,QvBzwe9B,CuBywEA,C;O;KAPX,C;iFAUA,yB;MAAA,2C;MvBzwEA,iD;MuBywEA,sC;QAOI,OAAO,qBvB3wEA,qBuB2weY,iBvB3weZ,EuB2wEsB,QAAS,QvB3we/B,CuB2wEA,C;O;KAPX,C;IAUA,2B;MAQI,IAAI,iBAAO,CAAX,C;QAAC,YAAU,SAAV,EAAGB,CAAhB,EAAmB,cAAAnB,C;K;IAGIB,2B;MAQI,IAAI,iBAAO,CAAX,C;QAAC,YAAU,SAAV,EAAGB,CAAhB,EAAmB,cAAAnB,C;K;IAGIB,2B;MAQI,IAAI,iBAAO,CAAX,C;QAAC,YAAU,SAAV,EAAGB,CAAhB,EAAmB,cAAAnB,C;K;IAGIB,+C;MAa0B,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MACzD,oCAAA,2BAAkB,SAaIB,EAA6B,OAA7B,EAAc,cAAAtC,C;MACb,YAAU,SAAV,EAAGB,SAAhB,EAA2B,OAA3B,C;K;IAGJ,+C;MAa2B,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MAC1D,oCAAA,2BAAkB,SAaIB,EAA6B,OAA7B,EAAc,cAAAtC,C;MACb,YAAU,SAAV,EAAGB,SAAhB,EAA2B,OAA3B,C;K;IAGJ,+C;MAa2B,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MAC1D,oCAAA,2BAAkB,SAaIB,EAA6B,OAA7B,EAAc,cAAAtC,C;MACb,YAAU,SAAV,EAAGB,SAAhB,EAA2B,OAA3B,C;K;IAGJ,+C;MAa4B,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MAC3D,oCAAA,2BAAkB,SAaIB,EAA6B,OAA7B,EAAc,cAAAtC,C;MACb,YAAU,SAAV,EAAGB,SAAhB,EAA2B,OAA3B,C;K;IAGJ,0D;MAaI,kBAAK,SAAL,EAAGB,OAAhB,C;MAh8CQ,WAAR,iBAAQ,EAI8CA,SAj8CA,EAI8CW,OAJ8CX,C;K;IAo8CZ,0D;MAaI,kBAAK,SAAL,EAAGB,OAAhB,C;MAj8CQ,WAAR,iBAAQ,EAK8CA,SAI8CA,EAK8CW,OAI8CX,C;K;IAq8CZ,0D;MAaI,kBAAK,SAAL,EAAGB,OAAhB,C;MAI8CQ,UAAR,iBAAQ,EAm8CA,SAN8CA,EAm8CW,OAN8CX,C;K;IAS8CZ,0D;MAaI,kBAAK,SAAL,EAAGB,OAAhB,C;MAN8CQ,WAAR,iBAAQ,EAo8CA,SAP8CA,EAo8CW,OP8CX,C;K;8FAu8CZ,qB;MAQI,OAAO,iBvB/jGiB,Q;K;4FuBkkG5B,qB;MAQI,OAAO,iBvBtjGiB,Q;K;8FuByjG5B,yB;MAAA,gD;MAAA,4B;QAQI,OAAE,OAAR,iBAAQ,C;O;KARnB,C;gGAWA,qB;MAQI,OAAO,iBvBtlGiB,Q;K;IuB+IGL,gD;MAAA,wB;QAAW,qCAAK,KAAL,C;O;K;IANIC,iC;MAMI,OAAO,iBAAM,cAAN,EAAY,8BAAZ,C;K;IASY,kD;MAAA,wB;QAAW,qCAAK,KAAL,C;O;K;IANIC,mC;MAMI,OAAO,iBAAM,cAAN,EAAY,gCAAZ,C;K;IASY,kD;MAAA,wB;QAAW,qCAAK,KAAL,C;O;K;IANIC,mC;MAMI,OAAO,iBAAM,cAAN,EAAY,gCAAZ,C;K;IASiB,gD;MAAA,wB;QAAW,yBAAK,KAAL,C;O;K;IANvC,iC;MAMI,OJxqIO,eAAW,+BIwqIA,gBJxqIA,GAAGB,kBIwqIV,8BJxqIU,CAAhB,CAAX,C;K;gGI2qIX,yB;MAAA,yC;MAAA,4B;QAQI,OAAO,oBAAW,SvBxpGM,QuBwpGjB,C;O;KARX,C;IAiB2B,8C;MAAA,wB;QAAW,wBAAK,KAAL,C;O;K;IANtC,gC;MAMI,OH5rIO,cAAU,gCG4rIA,gBH5rIA,GAAe,iBG4rIT,6BH5rIS,CAAF,CAAV,C;K;8FG+rIX,yB;MAAA,uC;MAAA,4B;QAQI,OAAO,mBAAU,SvBxpGO,QuBwpGjB,C;O;KARX,C;IAiB4B,gD;MAAA,wB;QAAW,yBAAK,KAAL,C;O;K;IANvC,iC;MAMI,OFhtIO,eAAW,kBEgtIA,gBFhtIA,EAAGB,kBEgtIV,8BFhtIU,CAAhB,CAAX,C;K;gGEmtIX,yB;MAAA,gD;MAAA,yC;MAAA,4B;QAQI,OAAO,oBAAgB,OAAL,SAAK,CAAhB,C;O;KARX,C;IAiB6B,kD;MAAA,wB;QAAW,0BAAK,KAAL,C;O;K;IANxC,kC;MAMI,ODpuIO,gBAAy,gCCouIA,gBDpuIA,GAAiB,mBCouIX,+BDpuIW,CAAjB,CAAZ,C;K;kGCuuIX,yB;MAAA,2C;MAAA,4B;QAQI,OAAO,qBAAy,SvB1sGK,QuB0sGjB,C;O;KARX,C;mGAWA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAcI,aAAa,mBAAyC,cAAIB,YAAy,cAAZ,CAAKB,EAAC,EAAD,CAAzC,C;QAsEG,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UArEO,MASEP,aAAI,OA AJ,EAtEe,aAsEF,CAAc,OAAd,CAAb,C;;QAtEhB,OA AuB,M;O;Kaf3B,C;mGakBA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAcI,aAAa,mBAA0C,cAAIB,YAAy,cAAZ,CAAKB,EAAC,EAAD,CAA1C,C;QAsEG,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UArEO,MASEP,aAAI,OA AJ,EAtEe,aAsEF,CAAc,OAAd,CAAb,C;;QAtEhB,OA AuB,M;O;Kaf3B,C;mGakBA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAcI,aAAa,mBAA2C,cAAIB,YAAy,cAAZ,CAAKB,EAAC,EAAD,CAA3C,C;QAsEG,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UArEO,MASEP,aAAI,OA AJ,EAtEe,aAsEF,CAAc,OAAd,CAAb,C;;QAtEhB,OA AuB,M;O;Kaf3B,C;uGakBA,iD;MAYoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,WAAY,aAAI,OA AJ,EA Aa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;uGAGX,iD;MAYoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,WAAY,aAAI,OA AJ,EA Aa,

cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;uGAGX,iD;MAYoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,WAAy,aAAI,OAaJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;uGAGX,iD;MAYoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,WAAy,aAAI,OAaJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;uFAGX,yB;MAAA,+D;MAoLA,gD;MApLA,uC;QASW,kBAAU,gB;QAkLD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAnL6B,SAmLIB,CAAU,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QApLhB,OAsLO,W;O;KA/LX,C;uFAYA,yB;MAAA,+D;MAsLA,gD;MatLA,uC;QASW,kBAAU,gB;QAoLD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WArL6B,SAqLIB,CAAU,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAtLhB,OAwLO,W;O;KAjMX,C;uFAYA,yB;MAAA,+D;MAwLA,gD;MAxLA,uC;QASW,kBAAU,gB;QAsLD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAvL6B,SAuLIB,CAAU,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAxLhB,OA0LO,W;O;KAnMX,C;uFAYA,yB;MAAA,+D;MA0LA,gD;MA1LA,uC;QASW,kBAAU,gB;QAwLD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAZL6B,SAyLIB,CAAU,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAIhB,OA4LO,W;O;KAjMX,C;qGAYAY,yB;MAAA,+D;MA4DA,gD;MA5DA,uC;QAYW,kBAAiB,gB;QA2DR,gB;QADhB,YAAy,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WA5DoC,SA4DzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAAnB,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QA7DhB,OA+DO,W;O;KA3EX,C;qGAeA,yB;MAAA,+D;MA+DA,gD;MA/DA,uC;QAYW,kBAAiB,gB;QA8DR,gB;QADhB,YAAy,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WA/DoC,SA+DzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAAnB,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QA9EhB,OAkEO,W;O;KA9EX,C;qGAeA,yB;MAAA,+D;MAkEA,gD;MAIEA,uC;QAYW,kBAAiB,gB;QAIER,gB;QADhB,YAAy,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAIeOC,SAkEzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAAnB,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAnEhB,OAqEO,W;O;KAjFX,C;qGAeA,yB;MAAA,+D;MAqEA,gD;MArEA,uC;QAYW,kBAAiB,gB;QAoER,gB;QADhB,YAAy,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WArEoC,SAqEzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAAnB,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAtEhB,OAweO,W;O;KApFX,C;yGAeA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAy,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAAnB,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;yGakBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAy,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAAnB,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;yGakBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAy,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAAnB,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;2FakBA,yB;MAAA,gD;MAAA,oD;QAOoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAXX,C;2FACa,yB;MAAA,gD;MAAA,oD;QAOoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAXX,C;2FACa,yB;MAAA,gD;MAAA,oD;QAOoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAXX,C;uFACa,yB;MAAA,wE;MA4HA,+D;MA5HA,yC;QAYW,kBAAU,oB;QA4HD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UA7HoD,WA6H1C,CAAY,OAAZ,C;UjC99IP,U;UADP,YiCg+Ie,WjCh+IH,WiCg+IwB,GjCh+IxB,C;UACL,IAAI,aAAJ,C;YACH,aiC89IuC,gB;YAA5B,WjC79IX,aiC69IgC,GjC79IhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiC09IA,iB;UACA,IAAK,WAAI,OAaJ,C;;QA/HT,OAIo,W;O;KA7IX,C;uFAeA,yB;MAAA,wE;MAiIA,+D;MAjIA,yC;QAYW,kBAAU,oB;QAIiD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAIqD,WAKI3C,CAAY,OAAZ,C;UjCI/IP,U;UADP,YiCo/Ie,WjCp/IH,WiCo/IwB,GjCp/IxB,C;UACL,IAAI,aAAJ,C;YACH,aiCk/IuC,gB;YAA5B,WjCj/IX,aiCi/IgC,GjCj/IhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiC8+IA,iB;UACA,IAAK,WAAI,OAaJ,C;;QApIT,OAsIO,W;O;KAIJX,C;sFAeA,yB;MAAA,wE;MAsIA,+D;MatIA,yC;QAYW,kBAAU,oB;QAsID,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAvIqD,WAuI3C,CAA

Y,OAAZ,C;UjCtgJP,U;UADP,YiCwgJe,WjCxgJH,WiCwgJwB,GjCxgJxB,C;UACL,IAAI,aAAJ,C;YACH,aiCsgJu  
C,gB;YAA5B,WjCrgJX,aiCqgJgC,GjCrgJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiCkgJA,iB;UACA,IAAK,WA  
AI,OAAJ,C;;QAzIT,OA2IO,W;O;KAvJX,C;uFAeA,yB;MAAA,wE;MA2IA,+D;MA3IA,yC;QAYW,kBAAU,oB;Q  
A2ID,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UA5IsD,WA4I5C,CAAY,OAAZ,C;UjC1hJP,U  
;UADP,YiC4hJe,WjC5hJH,WiC4hJwB,GjC5hJxB,C;UACL,IAAI,aAAJ,C;YACH,aiC0hJuC,gB;YAA5B,WjCzhJX  
,aiCyhJgC,GjCzhJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiCshJA,iB;UACA,IAAK,WAAI,OAAJ,C;;QA9IT,OA  
gJO,W;O;KA5JX,C;uFAeA,yB;MAAA,wE;MAgJA,+D;MAhJA,yD;QAaW,kBAAU,oB;QAqJD,Q;QAAA,2B;QA  
AhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UajJiD,WaiJvC,CAAY,OAAZ,C;UjCijJP,U;UADP,YiCijJe,WjCjjJH,  
WiCijJwB,GjCjjJxB,C;UACL,IAAI,aAAJ,C;YACH,aiC+iJuC,gB;YAA5B,WjC9iJX,aiC8iJgC,GjC9iJhC,EAAS,M  
AAT,C;YACA,e;;YAEA,c;;UiC2iJA,iB;UACA,IAAK,WAnJyD,cAmJrD,CAAE,OAAf,CAAJ,C;;QAnJT,OAqJO,  
W;O;KAIKX,C;uFAgBA,yB;MAAA,wE;MAqJA,+D;MArJA,yD;QAaW,kBAAU,oB;QAqJD,Q;QAAA,2B;QA  
AhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UatJiD,WAsJvC,CAAY,OAAZ,C;UjCpkJP,U;UADP,YiCskJe,WjCtkJH,  
WiCskJwB,GjCtkJxB,C;UACL,IAAI,aAAJ,C;YACH,aiCokJuC,gB;YAA5B,WjCnkJX,aiCmkJgC,GjCnkJhC,EA  
S,MAAT,C;YACA,e;;YAEA,c;;UiCgkJA,iB;UACA,IAAK,WAxJyD,cAwJrD,CAAE,OAAf,CAAJ,C;;QAxJT,OA0  
JO,W;O;KAvKX,C;uFAgBA,yB;MAAA,wE;MA0JA,+D;MA1JA,yD;QAaW,kBAAU,oB;QA0JD,Q;QAAA,2B;Q  
AAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UA3JiD,WA2JvC,CAAY,OAAZ,C;UjCzljJP,U;UADP,YiC2lJe,WjC3  
lJH,WiC2lJwB,GjC3lJxB,C;UACL,IAAI,aAAJ,C;YACH,aiCylJuC,gB;YAA5B,WjCxlJX,aiCwlJgC,GjCxlJhC,EA  
AS,MAAT,C;YACA,e;;YAEA,c;;UiCqlJA,iB;UACA,IAAK,WA7JyD,cA6JrD,CAAE,OAAf,CAAJ,C;;QA7JT,OA+  
JO,W;O;KA5KX,C;uFAgBA,yB;MAAA,wE;MA+JA,+D;MA/JA,yD;QAaW,kBAAU,oB;QA+JD,Q;QAAA,2B;Q  
AAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UAhKiD,WAgKvC,CAAY,OAAZ,C;UjC9mJP,U;UADP,YiCgnJe,W  
jChnJH,WiCgnJwB,GjChnJxB,C;UACL,IAAI,aAAJ,C;YACH,aiC8mJuC,gB;YAA5B,WjC7mJX,aiC6mJgC,GjC7  
mJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiC0mJA,iB;UACA,IAAK,WAlKyD,cAkKrD,CAAE,OAAf,CAAJ,C;;  
QAiKT,OAoKO,W;O;KAjLX,C;2FAgBA,yB;MAAA,+D;MAAA,sD;QAYoB,Q;QAAA,2B;QAAhB,OAAGB,cAA  
hB,C;UAAgB,yB;UACZ,UAAU,YAAy,OAAZ,C;UjC99IP,U;UADP,YiCg+Ie,WjCh+IH,WiCg+IwB,GjCh+IxhB,C;  
UACL,IAAI,aAAJ,C;YACH,aiC89IuC,gB;YAA5B,WjC79IX,aiC69IgC,GjC79IhC,EAAS,MAAT,C;YACA,e;;YA  
EA,c;;UiC09IA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAjBX,C;2FAoBA,yB;MAAA,+D;MAAA,  
sD;QAYoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAy,OAAZ,C;UjCI/IP,U;UAD  
P,YiCo/Ie,WjCp/IH,WiCo/IwB,GjCp/IxB,C;UACL,IAAI,aAAJ,C;YACH,aiCk/IuC,gB;YAA5B,WjCj/IX,aiCi/IgC,  
GjCj/IhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiC8+IA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;K  
AjBX,C;2FAoBA,yB;MAAA,+D;MAAA,sD;QAYoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAC  
Z,UAAU,YAAy,OAAZ,C;UjCtgJP,U;UADP,YiCwgJe,WjCxgJH,WiCwgJwB,GjCxgJxB,C;UACL,IAAI,aAAJ,C;  
YACH,aiCsgJuC,gB;YAA5B,WjCrgJX,aiCqgJgC,GjCrgJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiCkgJA,iB;UA  
CA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAjBX,C;2FAoBA,yB;MAAA,+D;MAAA,sD;QAYoB,Q;QAAA,  
2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAy,OAAZ,C;UjC1hJP,U;UADP,YiC4hJe,WjC5hJH  
,WiC4hJwB,GjC5hJxB,C;UACL,IAAI,aAAJ,C;YACH,aiC0hJuC,gB;YAA5B,WjCzhJX,aiCyhJgC,GjCzhJhC,EA  
S,MAAT,C;YACA,e;;YAEA,c;;UiCshJA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAjBX,C;2FAoB  
A,yB;MAAA,+D;MAAA,sE;QAaoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAy,  
OAAZ,C;UjCijJP,U;UADP,YiCijJe,WjCjjJH,WiCijJwB,GjCjjJxB,C;UACL,IAAI,aAAJ,C;YACH,aiC+iJuC,gB;YA  
A5B,WjC9iJX,aiC8iJgC,GjC9iJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiC2iJA,iB;UACA,IAAK,WAAI,eAAe,O  
AAf,CAAJ,C;;QAET,OAAO,W;O;KAIBX,C;2FAqBA,yB;MAAA,+D;MAAA,sE;QAaoB,Q;QAAA,2B;QAAhB,O  
AAGB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAy,OAAZ,C;UjCpkJP,U;UADP,YiCskJe,WjCtkJH,WiCskJwB,Gj  
CtkJxB,C;UACL,IAAI,aAAJ,C;YACH,aiCokJuC,gB;YAA5B,WjCnkJX,aiCmkJgC,GjCnkJhC,EAAS,MAAT,C;Y  
ACA,e;;YAEA,c;;UiCgkJA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAIBX,C;2FAqB  
A,yB;MAAA,+D;MAAA,sE;QAaoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAy,  
OAAZ,C;UjCzljJP,U;UADP,YiC2lJe,WjC3lJH,WiC2lJwB,GjC3lJxB,C;UACL,IAAI,aAAJ,C;YACH,aiCylJuC,gB;  
YAA5B,WjCxlJX,aiCwlJgC,GjCxlJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiCqlJA,iB;UACA,IAAK,WAAI,eA  
Ae,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAIBX,C;2FAqBA,yB;MAAA,+D;MAAA,sE;QAaoB,Q;QAAA,2B;QAA  
hB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAy,OAAZ,C;UjC9mJP,U;UADP,YiCgnJe,WjChnJH,WiCgn

JwB,GjChnJxB,C;UACL,IAAI,aAAJ,C;YACH,aiC8mJuC,gB;YAA5B,WjC7mJX,aiC6mJgC,GjC7mJhC,EAAS,M  
AAT,C;YACA,e;;YAEA,c;;UiC0mJA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KALBX,  
C;+EAqBA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAM,eAAa,cAAb,C;QAsKA,Q;QAAA,2B;QAAb,OAAa,cAAb,  
C;UAAa,sB;UACT,WAAy,WAvKiB,SAuKb,CAAU,IAAV,CAAJ,C;;QAvKhB,OAwKO,W;O;KAILX,C;+EAaA,  
yB;MAAA,gE;MAAA,uC;QAUW,kBAAM,eAAa,cAAb,C;QAsKA,Q;QAAA,2B;QAAb,OAAa,cAAb,C;UAAa,sB  
;UACT,WAAy,WAvKiB,SAuKb,CAAU,IAAV,CAAJ,C;;QAvKhB,OAwKO,W;O;KAILX,C;8EAaA,yB;MAAA,g  
E;MAAA,uC;QAUW,kBAAM,eAAa,cAAb,C;QAsKA,Q;QAAA,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WA  
AY,WAvKiB,SAuKb,CAAU,IAAV,CAAJ,C;;QAvKhB,OAwKO,W;O;KAILX,C;+EAaA,yB;MAAA,gE;MAAA,u  
C;QAUW,kBAAM,eAAa,cAAb,C;QAsKA,Q;QAAA,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAy,WAvKi  
B,SAuKb,CAAU,IAAV,CAAJ,C;;QAvKhB,OAwKO,W;O;KAILX,C;4FAaA,yB;MAAA,gE;MAAA,uC;QAUW,k  
BAAa,eAAa,cAAb,C;QAqDP,gB;QADb,YAAy,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAy,W  
AtDwB,SAsDpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QAtDhB,OAUdO,W;O;KAjEX,C;6FA  
aA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAa,eAAa,cAAb,C;QAwDP,gB;QADb,YAAy,C;QACC,2B;QAAb,OA  
Aa,cAAb,C;UAAa,sB;UACT,WAAy,WazDwB,SAyDpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,  
C;;QazDhB,OA0DO,W;O;KApEX,C;6FAaA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAa,eAAa,cAAb,C;QA2DP,g  
B;QADb,YAAy,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAy,WA5DwB,SA4DpB,EAAU,cAAV,  
EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QA5DhB,OA6DO,W;O;KAvEX,C;4FAaA,yB;MAAA,gE;MAAA,uC;  
QAUW,kBAAa,eAAa,cAAb,C;QA8DP,gB;QADb,YAAy,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,W  
AAy,WA/DwB,SA+DpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QA/DhB,OAgEO,W;O;KA1E  
X,C;iGAaA,6C;MAWiB,UACiB,M;MAF9B,YAAy,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAA  
Y,WAAI,WAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;MACHB,OAAO,W;K;iGAGX,6C;MAWiB,U  
ACiB,M;MAF9B,YAAy,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAy,WAAI,WAAU,cAAV,EA  
AU,sBAAV,WAAmB,IAAnB,CAAJ,C;;MACHB,OAAO,W;K;iGAGX,6C;MAWiB,UACiB,M;MAF9B,YAAy,C;  
MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAy,WAAI,WAAU,cAAV,EAAU,sBAAV,WAAmB,IAAn  
B,CAAJ,C;;MACHB,OAAO,W;K;iGAGX,6C;MAWiB,UACiB,M;MAF9B,YAAy,C;MACC,2B;MAAb,OAAa,cA  
Ab,C;QAAa,sB;QACT,WAAy,WAAI,WAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;MACHB,OAAO,  
W;K;mFAGX,6C;MAQiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAy,WAAI,UAAU,IAAV,C  
AAJ,C;;MACHB,OAAO,W;K;mFAGX,6C;MAQiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAy  
,WAAI,UAAU,IAAV,CAAJ,C;;MACHB,OAAO,W;K;mFAGX,6C;MAQiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;  
QAAa,sB;QACT,WAAy,WAAI,UAAU,IAAV,CAAJ,C;;MACHB,OAAO,W;K;mFAGX,6C;MAQiB,Q;MAAA,2B  
;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAy,WAAI,UAAU,IAAV,CAAJ,C;;MACHB,OAAO,W;K;IAUiB,6C;  
MAAA,mB;QAAE,gC;O;K;IAP9B,iC;MAOI,OAAO,qBAAiB,8BAAjB,C;K;IAUiB,6C;MAAA,mB;QAAE,gC;O;  
K;IAP9B,iC;MAOI,OAAO,qBAAiB,8BAAjB,C;K;IAUiB,6C;MAAA,mB;QAAE,gC;O;K;IAP9B,iC;MAOI,OAA  
O,qBAAiB,8BAAjB,C;K;IAUiB,6C;MAAA,mB;QAAE,gC;O;K;IAP9B,iC;MAOI,OAAO,qBAAiB,8BAAjB,C;K;  
+EAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,C  
AAL,C;UAAyB,OAAO,K;;MACTD,OAAO,I;K;+EAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;Q  
AAGB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,OAAO,K;;MACTD,OAAO,I;K;+EAGX,gC;MAS  
oB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,  
OAAO,K;;MACTD,OAAO,I;K;+EAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM  
,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,OAAO,K;;MACTD,OAAO,I;K;+EAGX,yB;MAAA,0C;MAAA,4B;  
QASI,OAAe,IAAR,iBAAQ,C;O;KATnB,C;+EAYA,yB;MAAA,0C;MAAA,4B;QASI,OAAe,IAAR,iBAAQ,C;O;K  
ATnB,C;+EAYA,yB;MAAA,0C;MAAA,4B;QASI,OAAe,IAAR,iBAAQ,C;O;KATnB,C;+EAYA,yB;MAAA,0C;M  
AAA,4B;QASI,OAAe,IAAR,iBAAQ,C;O;KATnB,C;+EAYA,gC;MASoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,  
C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,I;;MACrD,OAAO,K;K;+EAGX,gC;MASoB,  
Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,I;;M  
ACrD,OAAO,K;K;+EAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAA  
U,OAAV,CAAJ,C;UAAwB,OAAO,I;;MACrD,OAAO,K;K;+EAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAGB,c  
AAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,I;;MACrD,OAAO,K;K;mFAGX,gC;



MAQoB,Q;MADhB,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,C  
AAJ,C;UAAwB,qB;;MAC9C,OAAO,K;K;mFAGX,gC;MAQoB,Q;MADhB,YAAY,C;MACI,2B;MAAhB,OAAgB  
,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,qB;;MAC9C,OAAO,K;K;mFAGX,gC;MA  
QoB,Q;MADhB,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,  
C;UAAwB,qB;;MAC9C,OAAO,K;K;mFAGX,gC;MAQoB,Q;MADhB,YAAY,C;MACI,2B;MAAhB,OAAgB,cAA  
hB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,qB;;MAC9C,OAAO,K;K;iFAGX,yC;MAaoB,Q;  
MADhB,kBAakB,O;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,cAAc,UAAU,WAAV,EAAuB,O  
AAvB,C;;MACpC,OAAO,W;K;iFAGX,yC;MAaoB,Q;MADhB,kBAakB,O;MACF,2B;MAAhB,OAAgB,cAAhB,  
C;QAAgB,yB;QAAM,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;iFAGX,yC;MAaoB,Q;MA  
DhB,kBAakB,O;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,cAAc,UAAU,WAAV,EAAuB,OAA  
vB,C;;MACpC,OAAO,W;K;iFAGX,yC;MAaoB,Q;MADhB,kBAakB,O;MACF,2B;MAAhB,OAAgB,cAAhB,C;Q  
AAgB,yB;QAAM,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;+FAGX,yC;MAeoB,UAA8B,  
M;MAF9C,YAAY,C;MACZ,kBAakB,O;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,cAAc,WAA  
U,cAAV,EAAU,sBAAV,WAAmB,WAAAnB,EAAgC,OAAhC,C;;MACpC,OAAO,W;K;+FAGX,yC;MAeoB,UAA8  
B,M;MAF9C,YAAY,C;MACZ,kBAakB,O;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,cAAc,WAA  
AU,cAAV,EAAU,sBAAV,WAAmB,WAAAnB,EAAgC,OAAhC,C;;MACpC,OAAO,W;K;+FAGX,yC;MAeoB,UA  
A8B,M;MAF9C,YAAY,C;MACZ,kBAakB,O;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,cAAc,  
WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAAnB,EAAgC,OAAhC,C;;MACpC,OAAO,W;K;+FAGX,yC;MAeoB,  
UAA8B,M;MAF9C,YAAY,C;MACZ,kBAakB,O;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,cAA  
Ac,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAAnB,EAAgC,OAAhC,C;;MACpC,OAAO,W;K;0FAGX,yB;MA1  
uDI,8D;MA0uDJ,gD;QAeoC,Q;QAHhC,YAtvDgB,cAAR,iBAAQ,C;QAuvDhB,kBAakB,O;QACIB,OAAO,SAAS  
,CAAhB,C;UACI,cAAc,UAAU,uBAAI,YAAJ,EAAI,oBAAJ,QAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;K  
AjBX,C;2FAoBA,yB;MATvDI,8D;MASvDJ,gD;QAeoC,Q;QAHhC,YAlwDgB,cAAR,iBAAQ,C;QAmwDhB,kBAA  
kB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,YAAJ,EAAI,oBAAJ,QAAV,EAAwB,WAAx  
B,C;;QAEIB,OAAO,W;O;KAjBX,C;2FAoBA,yB;MA1wDI,8D;MAkwDJ,gD;QAeoC,Q;QAHhC,YA9wDgB,cAAR  
,iBAAQ,C;QA+wDhB,kBAakB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,YAAJ,EAAI,oB  
AAJ,QAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAjBX,C;2FAoBA,yB;MA9wDI,8D;MA8wDJ,gD;QAeoC  
,Q;QAHhC,YA1xDgB,cAAR,iBAAQ,C;QA2xDhB,kBAakB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UA  
AU,uBAAI,YAAJ,EAAI,oBAAJ,QAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAjBX,C;yGAoBA,yB;MA1z  
DI,8D;MA0zDJ,gD;QAaI,YAv0DgB,cAAR,iBAAQ,C;QAw0DhB,kBAakB,O;QACIB,OAAO,SAAS,CAAhB,C;U  
ACI,cAAc,UAAU,KAaV,EAAiB,sBAAI,KAaJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAnB  
X,C;yGAsBA,yB;MAx0DI,8D;MAw0DJ,gD;QAaI,YAr1DgB,cAAR,iBAAQ,C;QAs1DhB,kBAakB,O;QACIB,OA  
AO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAaV,EAAiB,sBAAI,KAaJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;Q  
AEJ,OAAO,W;O;KAnBX,C;yGAsBA,yB;MAT1DI,8D;MAS1DJ,gD;QAaI,YAn2DgB,cAAR,iBAAQ,C;QAo2DhB,  
kBAakB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAaV,EAAiB,sBAAI,KAaJ,CAAjB,EAA6B,  
WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAnBX,C;yGAsBA,yB;MAP2DI,8D;MAo2DJ,gD;QAaI,YAj3DgB,cA  
AR,iBAAQ,C;QAK3DhB,kBAakB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAaV,EAAiB,sBAA  
I,KAaJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAnBX,C;uFAsBA,6B;MAOoB,Q;MAAA,2B;  
MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,OAAO,OAAP,C;;K;uFAG1B,6B;MAOoB,Q;MAAA,2B;MAAhB,  
OAAgB,cAAhB,C;QAAgB,yB;QAAM,OAAO,OAAP,C;;K;uFAG1B,6B;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAh  
B,C;QAAgB,yB;QAAM,OAAO,OAAP,C;;K;qGAG1B,6B;MAUiB,UAAa,M;MAD1B,YAAY,C;MACC,2B;MAA  
b,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;K;qGAGvB,6B;MAUiB,UA  
Aa,M;MAD1B,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,sBAAP,WAA  
gB,IAAhB,C;;K;qGAGvB,6B;MAUiB,UAAa,M;MAD1B,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;  
QAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;K;qGAGvB,6B;MAUiB,UAAa,M;MAD1B,YAAY,C;M  
ACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;K;IAGvB,2B  
;MAyIB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OA1/DG,gBAA

R,iBAAQ,C;MA0/DhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IpC5wL8D,YoC4wL1D,GpC5wL2E,KAAjB,EoC4wLpD,CpC5wLiF,KAA7B,CoC4wL1D,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2B;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OArgEG,gBAR,iBAAQ,C;MAqgEhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,InBvxL+D,amBuxL3D,GnBvxL6E,KAAiB,EmBuxLrD,CnBvxLmF,KAA9B,CmBuxL3D,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2B;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OAhhEG,gBAR,iBAAQ,C;MAghEhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IrCl0L4E,0BqCk0LxE,GrCvIL8B,KAAI,GAAiB,GA3O8B,EqCk0LIE,CrCvILwB,KAAI,GAAiB,GA3O8B,CqCk0LxE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2B;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OA3hEG,gBAAR,iBAAQ,C;MA2hEhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,I nC70L6E,0BmC60LzE,GnCzmL8B,KAAI,GAAiB,KApO+B,EmC60LnE,CnCzmLwB,KAAI,GAAiB,KApO+B,CmC60LzE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;mFAGX,yB;MAAA,sE;MA1jEI,8D;MA0jEJ,sC;QAaI,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,cAAc,sBAAK,CAAL,C;QACd,gBAzkEgB,cAykEA,SAzKER,QAAQ,C;QA0kEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAGnB,OAAO,O;O;KA1BX,C;mFA6BA,yB;MAAA,sE;MA/kEI,8D;MA+kEJ,sC;QAaI,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,cAAc,sBAAK,CAAL,C;QACd,gBA9IEgB,cA8IEA,SA9IER,QAAQ,C;QA+IEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAGnB,OAAO,O;O;KA1BX,C;mFA6BA,yB;MAAA,sE;MApMEI,8D;MAomEJ,sC;QAaI,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,cAAc,sBAAK,CAAL,C;QACd,gBAnnEgB,cAmnEA,SAnnER,QAAQ,C;QAonEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAGnB,OAAO,O;O;KA1BX,C;mFA6BA,yB;MAAA,sE;MAznEI,8D;MAynEJ,sC;QAaI,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,cAAc,sBAAK,CAAL,C;QACd,gBAxoEgB,cAwoEA,SAxoER,QAAQ,C;QAyoEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAGnB,OAAO,O;O;KA1BX,C;+FA6BA,yB;MA9qEI,8D;MA8qEJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBAzrEgB,cAyrEA,SAzrER,QAAQ,C;QA0rEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAGnB,OAAO,O;O;KA1BX,C;+FAyBA,yB;MA/rEI,8D;MA+rEJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBA1sEgB,cA0sEA,SA1sER,QAAQ,C;QA2sEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAGnB,OAAO,O;O;KA1BX,C;+FAyBA,yB;MAhtEI,8D;MAgtEJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBA3tEgB,cA2tEA,SA3tER,QAAQ,C;QA4tEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAGnB,OAAO,O;O;KA1BX,C;kFAyBA,yB;MAAA,sE;MA1xEI,8D;MpBvwHJ,iB;MoByhMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAlYEG,cAAR,iBAAQ,C;QAKyEhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBniMG,MAAO,KoBmiMO,QpBniMP,EoBmiMiB,CpBniMjB,C;;QoBqiMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAjyEI,8D;MpB/wHJ,iB;MoBgjMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;U

AAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAjzEG,cAAR,iBAAQ,C;QAizEhB,AAU  
,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB1jMG,MAAO,KoB0jMO,QpB1jMP,EoB0j  
MiB,CpB1jMjB,C;;QoB4jMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAhzEI,8D;MpBvxHJ,iB;MoBuk  
MA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QAC  
F,OA0EhB,cAAR,iBAAQ,C;QAg0EhB,AAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,W  
pBj1MG,MAAO,KoBilMO,QpBj1MP,EoBilMiB,CpBj1MjB,C;;QoBmlMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;M  
AAA,sE;MA/zEI,8D;MpB/xHJ,iB;MoB8lMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,e  
AAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA/0EG,cAAR,iBAAQ,C;QA+0EhB,AAU,CAAV,iB;UACI,QAAQ,  
SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBxmMG,MAAO,KoBwmMO,QpBxmMP,EoBwmMiB,CpBxmMjB,C;;  
QoB0mMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA92EI,8D;MpBlxHJ,iB;MoBgoMA,sC;QAgBiB,Q;  
QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA93EG,cAAR,i  
BAAQ,C;QA83EhB,AAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB1oMG,MAAO,K  
oB0oMO,QpB1oMP,EoB0oMiB,CpB1oMjB,C;;QoB4oMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA73  
EI,8D;MpB1xHJ,iB;MoBupMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sB  
AAK,CAAL,CAAT,C;QACF,OA74EG,cAAR,iBAAQ,C;QA64EhB,AAU,CAAV,iB;UACI,QAAQ,SAAS,sBAA  
K,CAAL,CAAT,C;UACR,WpBjqMG,MAAO,KoBiqMO,QpBjqMP,EoBiqMiB,CpBjqMjB,C;;QoBmqMd,OAAO,  
Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA54EI,8D;MpBlyHJ,iB;MoB8qMA,sC;QAgBiB,Q;QAFb,IAAI,mBAA  
J,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA55EG,cAAR,iBAAQ,C;QA45EhB  
,AAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBxrMG,MAAO,KoBwrMO,QpBxrMP,  
EoBwrMiB,CpBxrMjB,C;;QoB0rMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA35EI,8D;MpB1yHJ,iB;  
MoBqsMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,  
C;QACF,OA36EG,cAAR,iBAAQ,C;QA26EhB,AAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;U  
ACR,WpB/sMG,MAAO,KoB+sMO,QpB/sMP,EoB+sMiB,CpB/sMjB,C;;QoBitMd,OAAO,Q;O;KApBX,C;mFAu  
BA,yB;MAAA,sE;MA18EI,8D;MA08EJ,sC;QAcBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,S  
AAS,sBAAK,CAAL,CAAT,C;QACF,OAx9EG,cAAR,iBAAQ,C;QAw9EhB,AAU,CAAV,iB;UACI,QAAQ,SA  
S,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApB  
X,C;mFAuBA,yB;MAAA,sE;MAz9EI,8D;MAy9EJ,sC;QAcBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACr  
B,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA+EG,cAAR,iBAAQ,C;QAu+EhB,AAU,CAAV,iB;UACI,QA  
AQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;  
O;KApBX,C;mFAuBA,yB;MAAA,sE;MAx+EI,8D;MAw+EJ,sC;QAcBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,  
6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA/EG,cAAR,iBAAQ,C;QAs/EhB,AAU,CAAV,iB;U  
ACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,O  
AAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAv/EI,8D;MAu/EJ,sC;QAcBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,M  
AAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAArgFG,cAAR,iBAAQ,C;QAqgFhB,AAU,CAA  
V,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QA  
GnB,OAAO,Q;O;KApBX,C;8FAuBA,yB;MAtiFI,8D;MpBvwHJ,iB;MoB6yMA,sC;QAcBiB,Q;QAFb,IAAI,mBAAJ  
,C;UAAe,OAAO,I;QActB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OApjFG,cAAR,iBAAQ,C;QAojFhB,AAU  
U,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBrzMG,MAAO,KoBqzMO,QpBrzMP,EoBq  
zMiB,CpBrzMjB,C;;QoBuzMd,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MANjFI,8D;MpB/wHJ,iB;MoBk0MA,sC;QAc  
iB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QActB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAjkFG,cAAR  
,iBAAQ,C;QAikFhB,AAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB10MG,MAAO,  
KoB00MO,QpB10MP,EoB00MiB,CpB10MjB,C;;QoB40Md,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MAhkFI,8D;Mp  
BvxHJ,iB;MoBu1MA,sC;QAcBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QActB,eAAe,SAAS,sBAAK,CAAL,C  
AAT,C;QACF,OA9kFG,cAAR,iBAAQ,C;QA8kFhB,AAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT  
,C;UACR,WpB/1MG,MAAO,KoB+1MO,QpB/1MP,EoB+1MiB,CpB/1MjB,C;;QoBi2Md,OAAO,Q;O;KAIBX,C;+  
FAqBA,yB;MA7kFI,8D;MpB/xHJ,iB;MoB42MA,sC;QAcBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QActB,eA  
Ae,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA3IFG,cAAR,iBAAQ,C;QA2IFhB,AAU,CAAV,iB;UACI,QAAQ,S  
AAS,sBAAK,CAAL,CAAT,C;UACR,WpBp3MG,MAAO,KoBo3MO,QpBp3MP,EoBo3MiB,CpBp3MjB,C;;QoBs

3Md,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MA1nFI,8D;MpBlxHJ,iB;MoB44MA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAxoFG,cAAR,iBAAQ,C;QAwoFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBp5MG,MAAO,KoBo5MO,QpBp5MP,EoBo5MiB,CpBp5MjB,C;;QoBs5Md,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MAvoFI,8D;MpB1xHJ,iB;MoBi6MA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OArpFG,cAAR,iBAAQ,C;QAqpFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBz6MG,MAAO,KoBy6MO,QpBz6MP,EoBy6MiB,CpBz6MjB,C;;QoB26Md,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MAppFI,8D;MpBlyHJ,iB;MoBs7MA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAiqFG,cAAR,iBAAQ,C;QAkqFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB97MG,MAAO,KoB87MO,QpB97MP,EoB87MiB,CpB97MjB,C;;QoBg8Md,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MAjqFI,8D;MpB1yHJ,iB;MoB28MA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA/qFG,cAAR,iBAAQ,C;QA+qFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBn9MG,MAAO,KoBm9MO,QpBn9MP,EoBm9MiB,CpBn9MjB,C;;QoBq9Md,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MA9sFI,8D;MA8sFJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA1tFG,cAAR,iBAAQ,C;QA0tFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MA3tFI,8D;MA2tFJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAvuFG,cAAR,iBAAQ,C;QAuuFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MAxuFI,8D;MAwuFJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OApyFG,cAAR,iBAAQ,C;QAovFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MArvFI,8D;MAqvFJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAjwFG,cAAR,iBAAQ,C;QAiwFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAIBX,C;2FAqBA,yB;MAAA,sE;MAlyFI,8D;MAkyFJ,kD;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAhzFG,cAAR,iBAAQ,C;QAgzFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;0FAuBA,yB;MAAA,sE;MAjzFI,8D;MAizFJ,kD;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA/zFG,cAAR,iBAAQ,C;QA+zFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;2FAuBA,yB;MAAA,sE;MAh0FI,8D;MAg0FJ,kD;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA90FG,cAAR,iBAAQ,C;QA80FhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;2FAuBA,yB;MAAA,sE;MA+0FI,8D;MA+0FJ,kD;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA71FG,cAAR,iBAAQ,C;QA61FhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;uGAuBA,yB;MA93FI,8D;MA83FJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA14FG,cAAR,iBAAQ,C;QA04FhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAIBX,C;sGAqBA,yB;MA34FI,8D;MA24FJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA5FG,cAAR,iBAAQ,C;QAu5FhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAIBX,C;uGAqBA,yB;MAx5FI,8D;MAw5FJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA6FG,cAAR,iBAAQ,C;QAo6FhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CA

AX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;uGAqBA,yB;MAR6FI,8D;MAq6FJ,kD;  
QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAj7FG,c  
AAR,iBAAQ,C;QAI7FhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,S  
AAQ,QAAR,EAAkB,CAAIb,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;IAqB  
A,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACTb,UAAU,sBAAK,CAAL,C;MACG,OA19FG,gBA  
AR,iBAAQ,C;MA09FhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IpC5uN8D,YoC4uN1D,GpC5u  
N2E,KAAjB,EoC4uNpD,CpC5uNiF,KAA7B,CoC4uN1D,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,i  
C;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACTb,UAAU,sBAAK,CAAL,C;MACG,OAj+FG,gBAAR,i  
BAAQ,C;MAi+FhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,InBnvN+D,amBmvN3D,GnBnvN6E  
,KAAIb,EmBmvNrD,CnBnvNmF,KAA9B,CmBmvN3D,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC  
;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACTb,UAAU,sBAAK,CAAL,C;MACG,OAx+FG,gBAAR,i  
BAAQ,C;MAw+FhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IrC1xN4E,0BqC0xNxE,GrC/iN8B,  
KAAL,GAAiB,GA308B,EqC0xNIE,CrC/iNwB,KAAL,GAAiB,GA308B,CqC0xNxE,IAAJ,C;UAAa,MAAM,C;;  
MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACTb,UAAU,sBAAK,CAAL,  
C;MACG,OA/+FG,gBAAR,iBAAQ,C;MA++FhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,InCjy  
N6E,0BmCiyNzE,GnC7jN8B,KAAL,GAAiB,KApO+B,EmCiyNnE,CnC7jNwB,KAAL,GAAiB,KApO+B,CmCiy  
NzE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,  
6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OA1hGG,gBAAR,iBAAQ,C;MA0hGhB,aAAU,CAAV,iB;QACI,QA  
AQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAA  
M,C;;MAE9C,OAAO,G;K;IAGX,2C;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAA  
K,CAAL,C;MACG,OArGG,gBAAR,iBAAQ,C;MAqiGhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QAC  
R,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;I  
AGX,2C;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OAhjG  
G,gBAAR,iBAAQ,C;MAgiGhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,G  
AAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,2C;MAYiB,Q;MA  
Fb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OA3jGG,gBAAR,iBAAQ,C;MA  
2jGhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX  
,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,  
OAAO,I;MACTb,UAAU,sBAAK,CAAL,C;MACG,OAlmGG,gBAAR,iBAAQ,C;MAkmGhB,aAAU,CAAV,iB;QA  
CI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,  
MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACTb,UAAU,sBA  
AK,CAAL,C;MACG,OAzmGG,gBAAR,iBAAQ,C;MAymGhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;  
QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,  
G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACTb,UAAU,sBAAK,CAAL,C;MACG,OAhn  
GG,gBAAR,iBAAQ,C;MAgnGhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,  
GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;M  
AFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACTb,UAAU,sBAAK,CAAL,C;MACG,OAvnGG,gBAAR,iBAAQ,C;MA  
unGhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX  
,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,2B;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,  
MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OAlqGG,gBAAR,iBAAQ,C;MAkqGhB,aAAU,CAAV,iB;Q  
ACI,QAAQ,sBAAK,CAAL,C;QACR,IpCp7N8D,YoCo7N1D,GpCp7N2E,KAAjB,EoCo7NpD,CpCp7NiF,KAA7B  
,CoCo7N1D,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2B;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,  
MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OA7qGG,gBAAR,iBAAQ,C;MA6qGhB,aAAU,CAAV,iB;  
QACI,QAAQ,sBAAK,CAAL,C;QACR,InB/7N+D,amB+7N3D,GnB/7N6E,KAAIb,EmB+7NrD,CnB/7NmF,KAA9  
B,CmB+7N3D,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2B;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAA  
e,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OAxrGG,gBAAR,iBAAQ,C;MAwrGhB,aAAU,CAAV,iB;  
QACI,QAAQ,sBAAK,CAAL,C;QACR,IrC1+N4E,0BqC0+NxE,GrC/vN8B,KAAL,GAAiB,GA308B,EqC0+NIE,C  
rC/vNwB,KAAL,GAAiB,GA308B,CqC0+NxE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2B;MAYiB

,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OAnsGG,gBAAR,iBAAQ ,C;MAmsGhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,InCr/N6E,0BmCq/NzE,GnCjxN8B,KAAL ,GAAiB,KApO+B,EmCq/NnE,CnCjxNwB,KAAL,GAAiB,KApO+B,CmCq/NzE,IAAJ,C;UAAa,MAAM,C;;MAEv B,OAAO,G;K;mFAGX,yB;MAAA,sE;MAluGI,8D;MAkuGJ,sC;QAaI,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB, cAAc,sBAAK,CAAL,C;QACd,gBAjvGgB,cAivGA,SAjvGR,QAAQ,C;QAKvGhB,IAAI,cAAa,CAAjB,C;UAAoB, OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;U ACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB, OAAO,O;O;KA1BX,C;mFA6BA,yB;MAAA,sE;MAvvGI,8D;MAuvGJ,sC;QAaI,IAAI,mBAAJ,C;UAAe,MAAM,6 B;QACrB,cAAc,sBAAK,CAAL,C;QACd,gBAtwGgB,cAswGA,SAtwGR,QAAQ,C;QAuwGhB,IAAI,cAAa,CAAj B,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAA K,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAA W,C;;;QAGnB,OAAO,O;O;KA1BX,C;mFA6BA,yB;MAAA,sE;MA5wGI,8D;MA4wGJ,sC;QAaI,IAAI,mBAAJ,C; UAAe,MAAM,6B;QACrB,cAAc,sBAAK,CAAL,C;QACd,gBA3xGgB,cA2xGA,SA3xGR,QAAQ,C;QA4xGhB,IA AI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI ,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C; YACV,WAAW,C;;;QAGnB,OAAO,O;O;KA1BX,C;mFA6BA,yB;MAAA,sE;MAjyGI,8D;MAiyGJ,sC;QAaI,IAAI, mBAAJ,C;UAAe,MAAM,6B;QACrB,cAAc,sBAAK,CAAL,C;QACd,gBAhzGgB,cAgzGA,SAhzGR,QAAQ,C;QA izGhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb ,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI, UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KA1BX,C;+FA6BA,yB;Mat1GI,8D;MA51GJ,sC;QASI,IAAI, mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBAj2GgB,cAi2GA,SAj2GR,QAAQ,C;QAK2G hB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M; UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UA AU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAtBX,C;+FAyBA,yB;MAv2GI,8D;MAu2GJ,sC;QASI,IAAI,mB AAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBAI3GgB,cAk3GA,SAI3GR,QAAQ,C;QAm3Gh B,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M; UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UA AU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAtBX,C;+FAyBA,yB;MAx3GI,8D;MAw3GJ,sC;QASI,IAAI,mB AAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBAn4GgB,cAm4GA,SAn4GR,QAAQ,C;QAo4Gh B,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M; UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UA AU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAtBX,C;+FAyBA,yB;MAz4GI,8D;MAy4GJ,sC;QASI,IAAI,mB AAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBAp5GgB,cAo5GA,SAP5GR,QAAQ,C;QAq5Gh B,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M; UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UA AU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAtBX,C;kFAyBA,yB;MAAA,sE;MA17GI,8D;MpBnjHJ,iB;MoB 6+NA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;Q ACF,OA18GG,cAAR,iBAAQ,C;QA08GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UAC R,WpBv/NG,MAAO,KoBu/NO,QpBv/NP,EoBu/NiB,CpBv/NjB,C;;;QoBy/Nd,OAAO,Q;O;KApBX,C;mFAuBA,yB ;MAAA,sE;MAz8GI,8D;MpB3jHJ,iB;MoBogOA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACr B,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAz9GG,cAAR,iBAAQ,C;QAY9GhB,aAAU,CAAV,iB;UACI,QA AQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB9gOG,MAAO,KoB8gOO,QpB9gOP,EoB8gOiB,CpB9gOjB,C;;;Q oBghOd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAx9GI,8D;MpBnkHJ,iB;MoB2hOA,sC;QAgBiB,Q;Q AFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAx+GG,cAAR,iB AAQ,C;QAw+GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBriOG,MAAO,Ko BqiOO,QpBriOP,EoBqiOiB,CpBriOjB,C;;;QoBuiOd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAv+GI,8D; MpB3kHJ,iB;MoBkjOA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,C AAL,CAAT,C;QACF,OA v/GG,cAAR,iBAAQ,C;QAu/GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,

CAAT,C;UACR,WpB5jOG,MAAO,KoB4jOO,QpB5jOP,EoB4jOiB,CpB5jOjB,C;;QoB8jOd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAthHI,8D;MpB9jHJ,iB;MoBolOA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAtiHG,cAAR,iBAAQ,C;QAsiHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB9IOG,MAAO,KoB8IOO,QpB9IOP,EoB8IOiB,CpB9IOjB,C;;QoBgmOd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAriHI,8D;MpBtkHJ,iB;MoB2mOA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OArjHG,cAAR,iBAAQ,C;QAqjHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBrnOG,MAAO,KoBqnOO,QpBrnOP,EoBqnOiB,CpBrnOjB,C;;QoBunOd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MApjHI,8D;MpB9kHJ,iB;MoBkoOA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OApkHG,cAAR,iBAAQ,C;QAokHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB5oOG,MAAO,KoB4oOO,QpB5oOP,EoB4oOiB,CpB5oOjB,C;;QoB8oOd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MANkHI,8D;MpBtlHJ,iB;MoBypOA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAnlHG,cAAR,iBAAQ,C;QAmIHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBnqOG,MAAO,KoBmqOO,QpBnqOP,EoBmqOiB,CpBnqOjB,C;;QoBqqOd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAlnHI,8D;MAknHJ,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAhOHG,cAAR,iBAAQ,C;QAgOHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAjoHI,8D;MAioHJ,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA/OHG,cAAR,iBAAQ,C;QA+oHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAhpHI,8D;MAgpHJ,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA9pHG,cAAR,iBAAQ,C;QA8pHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA/pHI,8D;MA+pHJ,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA7qHG,cAAR,iBAAQ,C;QA6qHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;8FAuBA,yB;MA9sHI,8D;MpBnjHJ,iB;MoBiwOA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA5tHG,cAAR,iBAAQ,C;QA4tHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBzwOG,MAAO,KoBywOO,QpBzwOP,EoBywOiB,CpBzwOjB,C;;QoB2wOd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA3tHI,8D;MpB3jHJ,iB;MoBsxOA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAzuHG,cAAR,iBAAQ,C;QAYuHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB9xOG,MAAO,KoB8xOO,QpB9xOP,EoB8xOiB,CpB9xOjB,C;;QoBgyOd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAXuHI,8D;MpBnkHJ,iB;MoB2yOA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAtvHG,cAAR,iBAAQ,C;QAsvHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBnzOG,MAAO,KoBmzOO,QpBnzOP,EoBmzOiB,CpBnzOjB,C;;QoBqzOd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MArVHI,8D;MpB3kHJ,iB;MoBg0OA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAnwHG,cAAR,iBAAQ,C;QAmwHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBx0OG,MAAO,KoBw0OO,QpBx0OP,EoBw0OiB,CpBx0OjB,C;;QoB00Od,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAlyHI,8D;MpB9jHJ,iB;MoBg2OA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAHzHG,cAAR,iBAAQ,C;QAgzHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBx2OG,MAAO,KoBw2OO,QpBx2OP,EoBw2OiB,CpBx2OjB,C;;QoB02Od,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA/yHI,8D;MpBtkHJ,iB;MoBq3OA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA7zHG,cAAR,iBAAQ,C;QA6zHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB73OG,MAAO,KoB63OO,QpB73OP,EoB63OiB,CpB73OjB,C;;QoB+3Od,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA5zHI,8D;MpB9kHJ,iB;MoB04OA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA10HG,cAA

R,iBAAQ,C;QA00HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB15OG,MAAO ,KoBk50O,QpB15OP,EoBk50iB,CpB15OjB,C;;QoBo5Od,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAz0HI,8D;MpBtl HJ,iB;MoB+5OA,sC;QAcIb,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAA T,C;QACF,OAv1HG,cAAR,iBAAQ,C;QAu1HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C; UACR,WpBv6OG,MAAO,KoBu6OO,QpBv6OP,EoBu60iB,CpBv60jB,C;;QoBy6Od,OAAO,Q;O;KAlBX,C;+FA qBA,yB;Mat3HI,8D;MA3HJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK, CAAL,CAAT,C;QACF,OAI4HG,cAAR,iBAAQ,C;QAK4HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CA AL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA, yB;MAN4HI,8D;MAM4HJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CA AL,CAAT,C;QACF,OA/4HG,cAAR,iBAAQ,C;QA+4HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL, CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB; MAh5HI,8D;MAG5HJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL, CAAT,C;QACF,OA55HG,cAAR,iBAAQ,C;QA45HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CA AT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA 75HI,8D;MA65HJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CA AT,C;QACF,OAz6HG,cAAR,iBAAQ,C;QAY6HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT, C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;2FAqBA,yB;MAAA, sE;MA18HI,8D;MA08HJ,kD;QAcIb,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,C AAL,CAAT,C;QACF,OAx9HG,cAAR,iBAAQ,C;QAw9HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAA L,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QA GnB,OAAO,Q;O;KApBX,C;0FAuBA,yB;MAAA,sE;MAz9HI,8D;MAy9HJ,kD;QAcIb,Q;QAFb,IAAI,mBAAJ,C; UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA+v+HG,cAAR,iBAAQ,C;QAu+HhB,a AAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAA I B,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;2FAuBA,yB;MAAA,sE;MAX+HI, 8D;MAw+HJ,kD;QAcIb,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT ,C;QACF,OA/HG,cAAR,iBAAQ,C;QAs/HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UA CR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q; O;KApBX,C;2FAuBA,yB;MAAA,sE;MAv/HI,8D;MAu/HJ,kD;QAcIb,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6 B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OArgIG,cAAR,iBAAQ,C;QAqgIhB,aAAU,CAAV,iB;UA CI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,C AAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;uGAuBA,yB;MATiII,8D;MASiIJ,kD;QAYiB,Q;QAFb, IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAljIG,cAAR,iBAAQ,C;Q AkjIhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAak B,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;SGAqBA,yB;MANjII,8D; MAmjIJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QA CF,OA/jIG,cAAR,iBAAQ,C;QA+jIhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IA AI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAl BX,C;uGAqBA,yB;MAhkII,8D;MAGkIJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SA AS,sBAAK,CAAL,CAAT,C;QACF,OA5kIG,cAAR,iBAAQ,C;QA4kIhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBA AK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAA W,C;;;QAGnB,OAAO,Q;O;KAlBX,C;uGAqBA,yB;MA7kII,8D;MA6kIJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UA Ae,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAzIIIG,cAAR,iBAAQ,C;QAYlIhB,aAAU,CAA V,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,G AakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;IAqBA,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C; QAe,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OAlOIG,gBAAR,iBAAQ,C;MAkoIhB,aAAU,CAAV,iB; QACI,QAAQ,sBAAK,CAAL,C;QACR,IpCp5P8D,YoCo5P1D,GpCp5P2E,KAAjB,EoCo5PpD,CpCp5PiF,KAA7B, CoCo5P1D,IAAJ,C;UAAa,MAAM,C;;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAe,O AAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OAzoIG,gBAAR,iBAAQ,C;MAyoIhB,aAAU,CAAV,iB;QACI,Q



AAQ,sBAAK,CAAL,C;QACR,InB35P+D,amB25P3D,GnB35P6E,KAAIB,EmB25PrD,CnB35PmF,KAA9B,CmB25P3D,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACTb,UAAU,sBAAK,CAAL,C;MACG,OAhpIG,gBAAR,iBAAQ,C;MAgpIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IrCl8P4E,0BqCk8PxE,GrCvtP8B,KAAAL,GAAiB,GA3O8B,EqCk8PIE,CrCvtPwB,KAAAL,GAAiB,GA3O8B,CqCk8PxE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACTb,UAAU,sBAAK,CAAL,C;MACG,OAvpIG,gBAAR,iBAAQ,C;MAupIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,InCz8P6E,0BmCy8PzE,GnCruP8B,KAAAL,GAAiB,KApO+B,EmCy8PnE,CnCruPwB,KAAAL,GAAiB,KApO+B,CmCy8PzE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OAlsIG,gBAAR,iBAAQ,C;MAksIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,2C;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OA7sIG,gBAAR,iBAAQ,C;MA6sIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,2C;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OAxtIG,gBAAR,iBAAQ,C;MAwtIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,2C;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OAnuIG,gBAAR,iBAAQ,C;MAmuhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACTb,UAAU,sBAAK,CAAL,C;MACG,OA1wIG,gBAAR,iBAAQ,C;MA0wIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACTb,UAAU,sBAAK,CAAL,C;MACG,OAjxIG,gBAAR,iBAAQ,C;MAixIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACTb,UAAU,sBAAK,CAAL,C;MACG,OAxxIG,gBAAR,iBAAQ,C;MAwxIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACTb,UAAU,sBAAK,CAAL,C;MACG,OA/xIG,gBAAR,iBAAQ,C;MA+xIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;iFAGX,qB;MASI,OAAO,mB;K;iFAGX,qB;MASI,OAAO,mB;K;iFAGX,qB;MASI,OAAO,mB;K;iFAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;qFAGX,6B;MAOmC,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QAAM,OAAO,OAAP,C;;MAArC,gB;K;qFAGJ,6B;MAOmC,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QAAM,OAAO,OAAP,C;;MAArC,gB;K;qFAGJ,6B;MAOmC,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QAAM,OAAO,OAAP,C;;MAArC,gB;K;qFAGJ,6B;MAOmC,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QAAM,OAAO,OAAP,C;;MAArC,gB;K;mGAGJ,6B;MATgFiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAghFnB,gB;K;mGAGJ,6B;MATgFiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAghFnB,gB;K;mGAGJ,6B;MATgFiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAghFnB,gB;K;qFAGJ,yB;MAAA,4F;MA9/II,8D;MA8/IJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,sBAAK,CAAL,C;QACD,OAjhJD,cAAR,iBAAQ,C;

QAihJhB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAtBX,C;qFAyBA,yB;MAAA,4F;MA/gJI,8D;MA+gJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,sBAAK,CAAL,C;QACD,OAliJD,cAAR,iBAAQ,C;QAkiJhB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAtBX,C;qFAyBA,yB;MAAA,4F;MAhiJI,8D;MAgiJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,sBAAK,CAAL,C;QACD,OAnjJD,cAAR,iBAAQ,C;QAmjJhB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAtBX,C;qFAyBA,yB;MAAA,4F;MAjjJI,8D;MAijJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,sBAAK,CAAL,C;QACD,OApkJD,cAAR,iBAAQ,C;QAokJhB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAtBX,C;mGAYBA,yB;MAAA,4F;MAImJI,8D;MAkmJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,sBAAK,CAAL,C;QACD,OAmJD,cAAR,iBAAQ,C;QAqnJhB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,KAAY,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;mGAYBA,yB;MAAA,4F;MAnnJI,8D;MAmJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,sBAAK,CAAL,C;QACD,OAtJD,cAAR,iBAAQ,C;QAsJhB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,KAAY,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;mGAYBA,yB;MAAA,4F;MApoJI,8D;MAooJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,sBAAK,CAAL,C;QACD,OAvpJD,cAAR,iBAAQ,C;QAupJhB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,KAAY,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;mGAYBA,yB;MAAA,4F;MArpJI,8D;MAqpJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,sBAAK,CAAL,C;QACD,OAxqJD,cAAR,iBAAQ,C;QAwqJhB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,KAAY,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;+GAYBA,yB;MAtsJI,8D;MAssJJ,uC;QAKBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAakB,sBAAK,CAAL,C;QACD,OAxTJD,cAAR,iBAAQ,C;QAwTJhB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,KAAY,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KArBX,C;+GAWBA,yB;MAttJI,8D;MAstJJ,uC;QAKBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAakB,sBAAK,CAAL,C;QACD,OAXuJD,cAAR,iBAAQ,C;QAWuJhB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,KAAY,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KArBX,C;+GAWBA,yB;MAtvJI,8D;MAsvJJ,uC;QAKBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAakB,sBAAK,CAAL,C;QACD,OAXvJD,cAAR,iBAAQ,C;QAWvJhB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,KAAY,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KArBX,C;+GAWBA,yB;MAuJI,8D;MAsuJJ,uC;QAKBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAakB,sBAAK,CAAL,C;QACD,OAXwJD,cAAR,iBAAQ,C;QAWwJhB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,KAAY,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KArBX,C;iGAWBA,yB;MAtyJI,8D;MAsyJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAakB,sBAAK,CAAL,C;QACD,OAZzJD,cAAR,iBAAQ,C;QAYzJhB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAtBX,C;iGAYBA,yB;MAvzJI,8D;MAuzJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAakB,sBAAK,CAAL,C;QACD,OA10JD,cAAR,iBAAQ,C;QA00JhB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAtBX,C;iGAYBA,yB;MAx0JI,8D;MAw0JJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAakB,sBAAK,CAAL,C;QACD,OA31JD,cAAR,iBAAQ,C;QA21JhB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAtBX,C;iGAYBA,yB;MAz1JI,8D;MAy1JJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAakB,sBAAK,CAAL,C;QACD,OA52JD,cAAR,iBAAQ,C;QA42JhB,iBAAC,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KAtBX,C;+FAYBA,yB;MAAA,4F;MA14JI,8D;MA04JJ,uC;QAKB0B,UAEU,M;QAJhC,YA15JgB,cAAR,iBAAQ,C;QA25JhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAakB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;+FAYBA,yB;MAAA,4F;MA35JI,8D;MA25JJ,uC;QAKB0B,UAEU,M;QAJhC,YA36JgB,cAAR,iBAAQ,C;QA46JhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAakB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,

UAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;+FAyBA,yB;MAAA,4F;MA56JI,8D;MA46JJ,uC;QA  
kB0B,UAEU,M;QAJhC,YA57JgB,cAAR,iBAAQ,C;QA67JhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+  
BAA9B,C;QACrB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU  
,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;+FAyBA,yB;MAAA,4F;  
MA77JI,8D;MA67JJ,uC;QakB0B,UAEU,M;QAJhC,YA78JgB,cAAR,iBAAQ,C;QA88JhB,IAAI,QAAQ,CAAZ,C;  
UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CA  
AhB,C;UACI,cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX  
,C;6GAyBA,yB;MAAA,4F;MA9+JI,8D;MA8+JJ,uC;QakB0B,Q;QAFtB,YA9/JgB,cAAR,iBAAQ,C;QA+/JhB,IA  
AI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACI  
B,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,  
qB;;QAEJ,OAAO,W;O;KAvBX,C;6GA0BA,yB;MAAA,4F;MAhgKI,8D;MaggKJ,uC;QakB0B,Q;QAFtB,YAhhK  
gB,cAAR,iBAAQ,C;QaihKhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAaKB,uB  
AAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,sBAAI,KAAJ,  
CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAvBX,C;6GA0BA,yB;MAAA,4F;MAlhKI,8D;MAkh  
KJ,uC;QakB0B,Q;QAFtB,YAliKgB,cAAR,iBAAQ,C;QAmiKhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B  
,+BAA9B,C;QACrB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAA  
U,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAvBX,C;6GA0BA,y  
B;MAAA,4F;MApiKI,8D;MAoiKJ,uC;QakB0B,Q;QAFtB,YApjKgB,cAAR,iBAAQ,C;QAqjKhB,IAAI,QAAQ,C  
AAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,S  
AAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,  
OAAO,W;O;KAvBX,C;yHA0BA,yB;MAtiKI,8D;MAsiKJ,uC;QaiB0B,Q;QAFtB,YArmKgB,cAAR,iBAAQ,C;QA  
smKhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,S  
AAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,  
OAAO,W;O;KAtBX,C;yHAyBA,yB;MAvmKI,8D;MAumKJ,uC;QaiB0B,Q;QAFtB,YAtmKgB,cAAR,iBAAQ,C;Q  
AunKhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,  
SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,  
OAAO,W;O;KAtBX,C;yHAyBA,yB;MAxnKI,8D;MAwnKJ,uC;QaiB0B,Q;QAFtB,YAvokgB,cAAR,iBAAQ,C;Q  
AwoKhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,  
SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,  
OAAO,W;O;KAtBX,C;yHAyBA,yB;MAzoKI,8D;MAyoKJ,uC;QaiB0B,Q;QAFtB,YAxpKgB,cAAR,iBAAQ,C;Q  
AypKhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,  
SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,  
OAAO,W;O;KAtBX,C;2GAyBA,yB;MA1rKI,8D;MA0rKJ,uC;QakB0B,UAEU,M;QAJhC,YA1sKgB,cAAR,iBA  
AQ,C;QA2sKhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,  
OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,O  
AAO,W;O;KAtBX,C;2GAyBA,yB;MA3sKI,8D;MA2sKJ,uC;QakB0B,UAEU,M;QAJhC,YA3tKgB,cAAR,iBAA  
Q,C;QA4tKhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,O  
AAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,OA  
AO,W;O;KAtBX,C;2GAyBA,yB;MA5tKI,8D;MA4tKJ,uC;QakB0B,UAEU,M;QAJhC,YA5uKgB,cAAR,iBAAQ,  
C;QA6uKhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OA  
AO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,OAA  
O,W;O;KAtBX,C;2GAyBA,yB;MA7uKI,8D;MA6uKJ,uC;QakB0B,UAEU,M;QAJhC,YA7vKgB,cAAR,iBAAQ,C  
;QA8vKhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAA  
O,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,  
W;O;KAtBX,C;+FAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QakBoB,Q;QAHhB,IAAI,mBAAJ,C;UAAe,OAA  
O,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arBziSO,W;QqB0iSP,kBA  
AkB,O;QACf,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd  
,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;+FAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QakBo

B,Q;QAHhB,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arBlkSO,W;QqBmkSP,kBAAkB,O;QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;+FAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QakBoB,Q;QAHhB,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arB3ISO,W;QqB4ISP,kBAAkB,O;QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;+FAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QakBoB,Q;QAHhB,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arBpnSO,W;QqBqnSP,kBAAkB,O;QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;6GAyBA,yB;MAAA,gD;MAAA,gE;MAI6KI,0D;MAk6KJ,gD;QAmBkB,gC;QAHd,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arB9oSO,W;QqB+oSP,kBAAkB,O;QACJ,OAr7KE,YAAR,iBAAQ,C;QAq7KF,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;6GA0BA,yB;MAAA,gD;MAAA,gE;MAp7KI,0D;MAo7KJ,gD;QAmBkB,gC;QAHd,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arBxqSO,W;QqByqSP,kBAAkB,O;QACJ,OA v8KE,YAAR,iBAAQ,C;QAu8KF,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;6GA0BA,yB;MAAA,gD;MAAA,gE;MAt8KI,0D;MA s8KJ,gD;QAmBkB,gC;QAHd,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arBlS SO,W;QqBmsSP,kBAAkB,O;QACJ,OA z9KE,YAAR,iBAAQ,C;QAy9KF,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;6GA0BA,yB;MAAA,gD;MAAA,gE;MAx9KI,0D;MAw9KJ,gD;QAmBkB,gC;QAHd,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arB5tSO,W;QqB6tSP,kBAAkB,O;QACJ,OA 3+KE,YAAR,iBAAQ,C;QA2+KF,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;mGA0BA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QakB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAIB,C;QACmC,kBAAtB,eAAgB,cAAhB,C;QAA8B,sBAAI,aAAJ,C;QAA3C,arBtvSO,W;QqBuvSe,qB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,aAAV,EAAuB,sBAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;mGayBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QakB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAIB,C;QACoC,kBAAvB,eAAiB,cAAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,arB/wSO,W;QqBgxSe,qB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,aAAV,EAAuB,sBAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;mGayBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QakB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAIB,C;QACqC,kBAAxB,eAAkB,cAAIB,C;QAAgC,sBAAI,aAAJ,C;QAA7C,arBj0SO,W;QqBk0Se,qB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,aAAV,EAAuB,sBAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;iHAYBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAmB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAIB,C;QACmC,kBAAtB,eAAgB,cAAhB,C;QAA8B,sBAAI,aAAJ,C;QAA3C,arB31SO,W;QqB41Se,qB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;iHA0BA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAmB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAIB,C;QACoC,kBAAvB,eAAiB,cAAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,arBr3SO,W;QqBs3Se,qB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;iHA0BA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAmB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,

OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAIB,C;QACoC,kBAAvB,eAAiB,cAAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,arB/4SO,W;QqBg5Se,qB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;iHA0BA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAmB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAIB,C;QACqC,kBAAxB,eAAkB,cAAIB,C;QAAgC,sBAAI,aAAJ,C;QAA7C,arBz6SO,W;QqB06Se,qB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;iFA0BA,yB;MAxZA,gD;MAAA,gE;MAwZA,gD;QAgBW,sB;;UAZS,Q;UAHhB,IAAI,mBAAJ,C;YAAe,qBAAO,OAYZH,OAZZG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBAwZzB,OAxZyB,C;UAA5C,arBziSO,W;UqB0iSP,kBAuZmB,O;UAtZH,2B;UAAhB,OAAGB,cAAhB,C;YAAgB,yB;YACZ,cAqZwB,SArZV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAKZP,yB;O;KAhBJ,C;iFAMBA,yB;MAIZA,gD;MAAA,gE;MAkZA,gD;QAgBW,sB;;UAhZS,Q;UAHhB,IAAI,mBAAJ,C;YAAe,qBAAO,OAmZH,OAnZG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBAkZzB,OAlZyB,C;UAA5C,arBlkSO,W;UqBmkSP,kBAiZmB,O;UAhZH,2B;UAAhB,OAAGB,cAAhB,C;YAAgB,yB;YACZ,cA+YwB,SA/YV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QA4YP,yB;O;KAhBJ,C;iFAMBA,yB;MA5YA,gD;MAAA,gE;MA4YA,gD;QAgBW,sB;;UA1YS,Q;UAHhB,IAAI,mBAAJ,C;YAAe,qBAAO,OA6YH,OA7YG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBA4YzB,OA5YyB,C;UAA5C,arB3lSO,W;UqB4lSP,kBA2YmB,O;UA1YH,2B;UAAhB,OAAGB,cAAhB,C;YAAgB,yB;YACZ,cAyYwB,SAzYV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAsYP,yB;O;KAhBJ,C;iFAMBA,yB;MAtYA,gD;MAAA,gE;MA5YA,gD;QAgBW,sB;;UApYS,Q;UAHhB,IAAI,mBAAJ,C;YAAe,qBAAO,OAuYH,OAvYG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBA5YzB,OAtYyB,C;UAA5C,arBpnSO,W;UqBqnSP,kBAqYmB,O;UApYH,2B;UAAhB,OAAGB,cAAhB,C;YAAgB,yB;YACZ,cAmYwB,SAnYV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAgYP,yB;O;KAhBJ,C;+FAMBA,yB;MAhYA,gD;MAAA,gE;MA16KI,0D;MAkyLJ,gD;QAiBW,6B;;UA9XO,gC;UAHd,IAAI,mBAAJ,C;YAAe,4BAAO,OAiYI,OAjYJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBAgYIB,OAhYkB,C;UAA5C,arB9oSO,W;UqB+oSP,kBA+X0B,O;UA9XZ,OA7KE,YAAR,iBAAQ,C;UAq7KF,mB;UAAA,kB;UAAA,kB;UAAAd,0D;YACI,cA6X+B,SA7XjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QA0XP,gC;O;KAjBJ,C;+FAoBA,yB;MA1XA,gD;MAAA,gE;MAp7KI,0D;MA8yLJ,gD;QAiBW,6B;;UAxXO,gC;UAHd,IAAI,mBAAJ,C;YAAe,4BAAO,OA2XI,OA3XJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBA0XIB,OA1XkB,C;UAA5C,arBxqSO,W;UqByqSP,kBAyX0B,O;UAxXZ,OA8KE,YAAR,iBAAQ,C;UAu8KF,mB;UAAA,kB;UAAA,kB;UAAAd,0D;YACI,cAuX+B,SAvXjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAoXP,gC;O;KAjBJ,C;+FAoBA,yB;MApXA,gD;MAAA,gE;MA8KI,0D;MA0zLJ,gD;QAiBW,6B;;UAIXO,gC;UAHd,IAAI,mBAAJ,C;YAAe,4BAAO,OAqXI,OA7XJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBAoXIB,OApxkB,C;UAA5C,arBlS0,W;UqBmsSP,kBAmX0B,O;UAIXZ,OA9KE,YAAR,iBAAQ,C;UAY9KF,mB;UAAA,kB;UAAA,kB;UAAAd,0D;YACI,cAiX+B,SAjXjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QA8WP,gC;O;KAjBJ,C;+FAoBA,yB;MA9WA,gD;MAAA,gE;MAx9KI,0D;MA80LJ,gD;QAiBW,6B;;UA5WO,gC;UAHd,IAAI,mBAAJ,C;YAAe,4BAAO,OA+Wl,OA/WJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBA8WIB,OA9WkB,C;UAA5C,arB5tSO,W;UqB6tSP,kBA6W0B,O;UA5WZ,OA3+KE,YAAR,iBAAQ,C;UA2+KF,mB;UAAA,kB;UAAA,kB;UAAAd,0D;YACI,cA2W+B,SA3WjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAwWP,gC;O;KAjBJ,C;mFAoBA,yB;MAAA,wB;MAAA,sC;QAuOB,Q;QADhB,UAGB,W;QACA,2B;QAaHb,OAAGB,cAAhB,C;UAGB,yB;UACZ,MpC3lTiD,SoC2lTjD,GpC3lT2D,KAAK,GoC2lTzD,SAAS,OAAT,CpC3lToE,KAAX,IAAf,C;;QoC6lTrD,OAAO,G;O;KAbX,C;mFAGBA,yB;MAAA,wB;MAAA,sC;QAuOB,Q;QADhB,UAGB,W;QACA,2B;QAaHb,OAAGB,cAAhB,C;UAGB,yB;UACZ,MpC3mTiD,SoC2mTjD,GpC3mT2D,KAAK,GoC2mTzD,SAAS,OAAT,CpC3mToE,KAAX,IAAf,C;;QoC6mTrD,OAAO,G;O;KAbX,C;mFAGBA,yB;MAAA,wB;MAAA,sC;QAuOB,Q;QADhB,UAGB,W;QACA,2B;QAaHb,OAAGB,cAAhB,C;UAGB,yB;UACZ,MpC3nTiD,SoC2nTjD,GpC3nT2D,KAAK,GoC2nTzD,SAAS,OAAT,CpC3nTo

E,KAAX,IAAf,C;;QoC6nTrD,OAAO,G;O;KAbX,C;mFAGBA,yB;MAAA,wB;MAAA,sC;QAUoB,Q;QADhB,UA  
AgB,W;QACA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,MpC3oTiD,SoC2oTjD,GpC3oT2D,KAAK,Go  
C2oTzD,SAAS,OAAT,CpC3oToE,KAAX,IAAf,C;;QoC6oTrD,OAAO,G;O;KAbX,C;8FAGBA,+B;MAUoB,Q;MA  
DhB,UAAkB,G;MACF,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAA  
O,G;K;+FAGX,+B;MAUoB,Q;MADhB,UAAkB,G;MACF,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QACZ,OA  
AO,SAAS,OAAT,C;;MAEX,OAAO,G;K;+FAGX,+B;MAUoB,Q;MADhB,UAAkB,G;MACF,2B;MAAhB,OAAG  
B,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;+FAGX,+B;MAUoB,Q;MADhB,UA  
AkB,G;MACF,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;k  
FAGX,+B;MAYoB,Q;MADhB,UAAoB,C;MACJ,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAA  
S,OAAT,C;;MAEX,OAAO,G;K;mFAGX,+B;MAYoB,Q;MADhB,UAAoB,C;MACJ,2B;MAAhB,OAAGB,cAAhB,  
C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;mFAGX,+B;MAYoB,Q;MADhB,UAAoB,C;M  
ACJ,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;mFAGX,+B  
;MAYoB,Q;MADhB,UAAoB,C;MACJ,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,  
C;;MAEX,OAAO,G;K;mFAGX,+B;MAYoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAGB,cAAhB,C;QAAgB  
,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAYoB,Q;MADhB,UAAe,C;MACC,2  
B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;  
MAYoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QACZ,YAAO,SAAS,OAAT,C  
AAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAYoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAGB,cAAhB,C;QA  
AgB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,yB;MAAA,SAWoB,gB;MAXpB,sC;Q  
AYoB,Q;QADhB,Y;QACgB,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;Q  
AEJ,OAAO,G;O;KAFX,C;mFAkBA,yB;MAAA,SAWoB,gB;MAXpB,sC;QAYoB,Q;QADhB,Y;QACgB,2B;QAAh  
B,OAAGB,cAAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAFX,C;mFAkBA,yB;  
MAAA,SAWoB,gB;MAXpB,sC;QAYoB,Q;QADhB,Y;QACgB,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ  
,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAFX,C;mFAkBA,yB;MAAA,SAWoB,gB;MAXpB,sC;QAYo  
B,Q;QADhB,Y;QACgB,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,  
OAAO,G;O;KAFX,C;mFAkBA,yB;MpChnTA,6B;MoCgnTA,sC;QAaoB,Q;QADhB,UpClnTmC,coCknTnB,CpCln  
TmB,C;QoCmnTnB,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,MpCt7TiD,coCs7TjD,GpCt7T2D,KAAK,  
GoCs7TzD,SAAS,OAAT,CpCt7ToE,KAAX,IAAf,C;;QoCw7TrD,OAAO,G;O;KAhBX,C;mFAMBA,yB;MpCnoT  
A,6B;MoCmoTA,sC;QAaoB,Q;QADhB,UpCroTmC,coCqoTnB,CpCroTmB,C;QoCsoTnB,2B;QAAhB,OAAGB,cA  
AhB,C;UAAgB,yB;UACZ,MpCz8TiD,coCy8TjD,GpCz8T2D,KAAK,GoCy8TzD,SAAS,OAAT,CpCz8ToE,KAA  
X,IAAf,C;;QoC28TrD,OAAO,G;O;KAhBX,C;mFAMBA,yB;MpCtpTA,6B;MoCspTA,sC;QAaoB,Q;QADhB,UpC  
xpTmC,coCwpTnB,CpCxpTmB,C;QoCypTnB,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,MpC59TiD,coC  
49TjD,GpC59T2D,KAAK,GoC49TzD,SAAS,OAAT,CpC59ToE,KAAX,IAAf,C;;QoC89TrD,OAAO,G;O;KAhBX,  
C;mFAMBA,yB;MpCzqTA,6B;MoCyqTA,sC;QAaoB,Q;QADhB,UpC3qTmC,coC2qTnB,CpC3qTmB,C;QoC4qTn  
B,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,MpC/+TiD,coC++TjD,GpC/+T2D,KAAK,GoC++TzD,SAAS  
,OAAT,CpC/+ToE,KAAX,IAAf,C;;QoCi/TrD,OAAO,G;O;KAhBX,C;mFAMBA,yB;MnBzrTA,+B;MmByrTA,sC;  
QAaoB,Q;QADhB,UnB1rTqC,eAAW,oBmB0rT/B,CnB1rT+B,CAAX,C;QmB2rTrB,2B;QAAhB,OAAGB,cAAhB,  
C;UAAgB,yB;UACZ,MnB//TmD,emB+/TnD,GnB//T8D,KAAK,KmB+/T5D,SAAS,OAAT,CnB//TuE,KAAX,CA  
AhB,C;;QmBigUvD,OAAO,G;O;KAhBX,C;mFAMBA,yB;MnB5sTA,+B;MmB4sTA,sC;QAaoB,Q;QADhB,UnB7  
sTqC,eAAW,oBmB6sT/B,CnB7sT+B,CAAX,C;QmB8sTrB,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,M  
nBlhUmD,emBkhUnD,GnBlhU8D,KAAK,KmBkhU5D,SAAS,OAAT,CnBlhUuE,KAAX,CAAhB,C;;QmBohUvD,  
OAAO,G;O;KAhBX,C;mFAMBA,yB;MnB/tTA,+B;MmB+tTA,sC;QAaoB,Q;QADhB,UnBhuTqC,eAAW,oBmBg  
uT/B,CnBhuT+B,CAAX,C;QmBiuTrB,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,MnBriUmD,emBqiUnD  
,GnBriU8D,KAAK,KmBqiU5D,SAAS,OAAT,CnBriUuE,KAAX,CAAhB,C;;QmBuiUvD,OAAO,G;O;KAhBX,C;  
mFAMBA,yB;MnBlvTA,+B;MmBkvTA,sC;QAaoB,Q;QADhB,UnBnvTqC,eAAW,oBmBmvT/B,CnBnvT+B,CAA  
X,C;QmBovTrB,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,MnBxjUmD,emBwjUnD,GnBxjU8D,KAAK,  
KmBwjU5D,SAAS,OAAT,CnBxjUuE,KAAX,CAAhB,C;;QmB0jUvD,OAAO,G;O;KAhBX,C;IAmBA,kC;MA2DI  
,WpBv8TO,MAAO,KoBu8TG,cpBv8TH,EoBq5TH,KAkDkB,OpBv8Tf,C;MoBw8Td,WAAW,iBAAa,IAAb,C;MA

CX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WArDqB,GAqDP,sBAAK,CAAL,CArDO,EAAAnB,KAqDqB,C  
AAM,CAAN,CArDF,CAqDrB,C;;MArDT,OAUdO,I;K;IApDX,kC;MAkEI,WpB19TO,MAAO,KoB09TG,cpB19T  
H,EoBi6TH,KAYdKB,OpB19Tf,C;MoB29Td,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;Q  
ACI,IAAK,WA5DqB,GA4DP,sBAAK,CAAL,CA5DO,EAAAnB,KA4DqB,CAAM,CAAN,CA5DF,CA4DrB,C;;MA  
5DT,OA8DO,I;K;IA3DX,kC;MAyEI,WpB7+TO,MAAO,KoB6+TG,cpB7+TH,EoB66TH,KAgEkB,OpB7+Tf,C;M  
oB8+Td,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAnEqB,GAmEP,sBAA  
K,CAAL,CAnEO,EAAAnB,KAmEqB,CAAM,CAAN,CAnEF,CAmErB,C;;MAnET,OAqEO,I;K;IAIEX,kC;MAgFI,  
WpBhgUO,MAAO,KoBggUG,cpBhgUH,EoBy7TH,KAuEkB,OpBhgUf,C;MoBigUd,WAAW,iBAaA,IAAb,C;MA  
CX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA1EqB,GA0EP,sBAAK,CAAL,CA1EO,EAAAnB,KA0EqB,C  
AAM,CAAN,CA1EF,CA0ErB,C;;MA1ET,OA4EO,I;K;+EAzEX,yB;MAAA,gE;MpB18TA,iB;MoBk8TA,8C;QA  
WI,WpBv8TO,MAAO,KoBu8TG,cpBv8TH,EoBu8TS,KAAM,OpBv8Tf,C;QoBw8Td,WAAW,eAAa,IAAb,C;QA  
CX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAMB,MAAM,CAAN  
,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpBr9TA,iB;MoBq9TA,8C;QAWI,Wp  
B19TO,MAAO,KoB09TG,cpB19TH,EoB09TS,KAAM,OpB19Tf,C;QoB29Td,WAAW,eAAa,IAAb,C;QACX,aAA  
U,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB  
,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpBx+TA,iB;MoBw+TA,8C;QAWI,WpB7+TO  
,MAAO,KoB6+TG,cpB7+TH,EoB6+TS,KAAM,OpB7+Tf,C;QoB8+Td,WAAW,eAAa,IAAb,C;QACX,aAAU,CA  
AV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAA  
J,C;;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpB3/TA,iB;MoB2/TA,8C;QAWI,WpBhgUO,MAA  
O,KoBggUG,cpBhgUH,EoBggUS,KAAM,OpBhgUf,C;QoBigUd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,M  
AAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;Q  
AET,OAAO,I;O;KAhBX,C;IAmBA,kC;MA8DoB,gB;MAHhB,gBAAgB,c;MACHb,WAAW,iBpBpkUJ,MAAO,K  
oBokUsB,wBAnDzB,KAmDyB,EAAwB,EAAXB,CpBpkUtB,EoBokUmD,SpBpkUnD,CoBokUH,C;MACX,QAA  
Q,C;MACQ,OArdL,KAqDK,W;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K  
;QACpB,IAAK,WAvDqB,GAuDP,uBAAK,UAAL,EAAK,kBAAL,UAvDO,EAuDI,OAxDJ,CAuDrB,C;;MAvDT,  
OAyDO,I;K;IAtdX,kC;MAuEoB,gB;MAHhB,gBAAGB,c;MACHb,WAAW,iBpBzIUJ,MAAO,KoBylUsB,wBA5D  
zB,KA4DyB,EAAwB,EAAXB,CpBzIUtB,EoBylUmD,SpBzIUUnD,CoBylUH,C;MACX,QAAQ,C;MACQ,OA9DL,  
KA8DK,W;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAh  
EqB,GAgEP,uBAAK,UAAL,EAAK,kBAAL,UAhEO,EAgEI,OAHEJ,CAGErB,C;;MAhET,OAkEO,I;K;IA/DX,kC;  
MAGFoB,gB;MAHhB,gBAAGB,c;MACHb,WAAW,iBpB9mUJ,MAAO,KoB8mUsB,wBArEzB,KAqEyB,EAAwB,  
EAAXB,CpB9mUtB,EoB8mUmD,SpB9mUnD,CoB8mUH,C;MACX,QAAQ,C;MACQ,OAveL,KAuEK,W;MAAh  
B,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAZEqB,GAYEP,uBA  
AK,UAAL,EAAK,kBAAL,UAZEO,EAYEI,OAzEJ,CAYErB,C;;MAzET,OA2EO,I;K;IAxEX,kC;MAyFoB,gB;MA  
HhB,gBAAGB,c;MACHb,WAAW,iBpBnoUJ,MAAO,KoBmoUsB,wBA9EzB,KA8EyB,EAAwB,EAAXB,CpBnoU  
tB,EoBmoUmD,SpBnoUnD,CoBmoUH,C;MACX,QAAQ,C;MACQ,OAhFL,KAGFK,W;MAAhB,OAAGB,cAAhB,  
C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAlFqB,GAKFP,uBAAK,UAAL,EAAK,k  
BAAL,UAIFO,EAKFI,OAIFJ,CAkFrB,C;;MAIFT,OAoFO,I;K;+EAjFX,yB;MAAA,kF;MAAA,gE;MpB9jUA,iB;M  
oB8jUA,8C;QAcOB,UAEY,M;QAL5B,gBAAGB,c;QChB,WAAW,epBpkUJ,MAAO,KoBokUsB,wBAAN,KAA  
M,EAAwB,EAAXB,CpBpkUtB,EoBokUmD,SpBpkUnD,CoBokUH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAG  
B,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,uBAAK,UAAL,E  
AAK,kBAAL,UAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KAIBX,C;+EAqBA,yB;MAAA,kF;MAAA,gE;  
MpBnlUA,iB;MoBmlUA,8C;QAcOB,UAEY,M;QAL5B,gBAAGB,c;QChB,WAAW,epBzIUJ,MAAO,KoBylUsB,  
wBAAN,KAAAM,EAAwB,EAAXB,CpBzIUtB,EoBylUmD,SpBzIUUnD,CoBylUH,C;QACX,QAAQ,C;QACQ,uB;Q  
AAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,uB  
AAK,UAAL,EAAK,kBAAL,UAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KAIBX,C;+EAqBA,yB;MAAA,  
kF;MAAA,gE;MpBxmUA,iB;MoBwmUA,8C;QAcOB,UAEY,M;QAL5B,gBAAGB,c;QChB,WAAW,epB9mUJ,  
MAAO,KoB8mUsB,wBAAN,KAAAM,EAAwB,EAAXB,CpB9mUtB,EoB8mUmD,SpB9mUnD,CoB8mUH,C;QAC  
X,QAAQ,C;QACQ,uB;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,I

AAK,WAAI,UAAU,uBAAK,UAAAL,EAAK,kBAAL,UAAV,EAAqB,OAArB,CAAJ,C;;QAET,OAAO,I;O;KAIBX,C;8EAqBA,yB;MAAA,kF;MAAA,gE;MpB7nUA,iB;MoB6nUA,8C;QAcoB,UAEY,M;QAL5B,gBAAgB,c;QACHB,WAAW,epBnoUJ,MAAO,KoBmoUsB,wBAAN,KAAM,EAawB,EAaxB,CpBnoUtB,EoBmoUmD,SpBnoUnD,CoBmoUH,C;QACX,QAAQ,C;QACQ,uB;QAahB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,uBAAK,UAAAL,EAAK,kBAAL,UAAV,EAAqB,OAArB,CAAJ,C;;QAET,OAAO,I;O;KAIBX,C;IAqBA,kC;MA2DI,WpBvsUO,MAAO,KoBusUG,cpBvsUH,EoBqpUH,KAKdKB,KpBvsUf,C;MoBwsUd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WArDqB,GAqDP,sBAAK,CAAL,CArDO,EAAAnB,KArDqB,aAAM,CAAN,CArDF,CAqDrB,C;;MArDT,OAUdO,I;K;IApDX,kC;MAkEI,WpB1tUO,MAAO,KoB0tUG,cpB1tUH,EoBiqUH,KAyDkB,KpB1tUf,C;MoB2tUd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA5DqB,GA4DP,sBAAK,CAAL,CA5DO,EAAAnB,KA4DqB,aAAM,CAAN,CA5DF,CA4DrB,C;;MA5DT,OA8DO,I;K;IA3DX,kC;MAyEI,WpB7uUO,MAAO,KoB6uUG,cpB7uUH,EoB6qUH,KAgEkB,KpB7uUf,C;MoB8uUd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAnEqB,GAmEP,sBAAK,CAAL,CAnEO,EAAAnB,KAmEqB,aAAM,CAAN,CAnEF,CAMerB,C;;MANET,OAqEO,I;K;IAIEX,kC;MAGFI,WpBhwUO,MAAO,KoBgwUG,cpBhwUH,EoByrUH,KAUekB,KpBhwUf,C;MoBiwUd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA1EqB,GA0EP,sBAAK,CAAL,CA1EO,EAAAnB,KA0EqB,aAAM,CAAN,CA1EF,CA0ErB,C;;MA1ET,OA4EO,I;K;+EAzEX,yB;MAAA,gE;MpBlSUA,iB;MoBksUA,8C;QAWI,WpBvsUO,MAAO,KoBusUG,cpBvsUH,EoBusUS,KAAM,KpBvsUf,C;QoBwsUd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,kBAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpBrTUA,iB;MoBqtUA,8C;QAWI,WpB1tUO,MAAO,KoB0tUG,cpB1tUH,EoB0tUS,KAAM,KpB1tUf,C;QoB2tUd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,kBAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpBxuUA,iB;MoBwuUA,8C;QAWI,WpB7uUO,MAAO,KoB6uUG,cpB7uUH,EoB6uUS,KAAM,KpB7uUf,C;QoB8uUd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,kBAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpB3vUA,iB;MoB2vUA,8C;QAWI,WpBhwUO,MAAO,KoBgwUG,cpBhwUH,EoBgwUS,KAAM,KpBhwUf,C;QoBiwUd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,kBAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;IAmBA,2B;MAQoB,Q;MADhB,UAAgB,W;MACHB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,MpCr8UiD,SoCq8UjD,GpCr8U2D,KAAK,GoCq8UzD,OpCr8UoE,KAAX,IAAf,C;;MoCu8UrD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAiB,2B;MACjB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,MnBh9UmD,UmBg9UnD,GnBh9U8D,KAAK,KmBg9U5D,OnBh9UuE,KAAX,CAAhB,C;;MmBk9UvD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAgB,W;MACHB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,MpCj+UiD,SoCi+UjD,GpCj+U2D,KAAK,GAAW,CD2O5C,SqCsvUxB,OrCtvUkC,KAAL,GAAiB,GAAtB,CC3O4C,MAAX,IAAf,C;;MoCm+UrD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAgB,W;MACHB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,MpC/+UiD,SoC++UjD,GpC/+U2D,KAAK,GAAW,CC4O5C,SmCmwUxB,OnCnwUkC,KAAL,GAAiB,KAAtB,CD5O4C,MAAX,IAAf,C;;MoCi+UrD,OAAO,G;K;+EAGX,yB;MAAA,0C;MpC5rUA,6B;MoC4rUA,4B;QAOI,OpCzrUmC,coCyrUpB,IAAR,iBAAQ,CpCzrUoB,C;O;KoCkrUvC,C;+EAUA,yB;MAAA,0C;MnBvrUA,+B;MmBurUA,4B;QAOI,OnBprUsC,emBorUvB,IAAR,iBAAQ,CnBprUuB,C;O;KmB6qU1C,C;+EAUA,yB;MAAA,sC;MpChtUA,6B;MoCgtUA,iBAOiB,yB;QrC7yUb,6B;eqC6yUa,c;UAAE,OrCpyUoB,cqCoyUpB,ErCpyU8B,KAAL,GAAiB,GAAtB,C;S;OqCoyUtB,C;MAPjB,4B;QA7iBoB,Q;QADhB,UpCxpTmC,coCwpTnB,CpCxpTmB,C;QoCypTnB,2B;QAahB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,MpC59TiD,coC49TjD,GpC59T2D,KAAK,GAAW,CD2O5C,cqCivTf,OrCjvTyB,KAAL,GAAiB,GAAtB,CC3O4C,MAAX,IAAf,C;;QoC+gVrD,OAjjiBO,G;O;KA0iBX,C;+EAUA,yB;MAAA,sC;MpC1tUA,6B;MoC0tUA,iBAOiB,yB;QnCtzUb,6B;emCszUa,c;UAAE,OnC7yUoB,cmC6yUpB,EnC7yU8B,KAAL,GAAiB,KAAtB,C;S;OmC6yUtB,C;MAPjB,4B;QApiBoB,Q;QADhB,UpC3qTmC,coC2qTnB,CpC3qTmB,C;QoC4qTnB,2B;QAahB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,MpC/+TiD,coC++TjD,GpC/+T2D,KAAK,GAAW,CC4O5C,cmCmwTf,OnCnwTyB,KAAL,GAAiB,KAAtB,CD5O4C,MAAX,IAAf,C;;QoCyhVrD,OAXiBO,G;O;KAiiBX,C;IC/kVA,mC;MAQoB,UACL,M;MAHX,aAAa,gBAAW,cAAX,C;MACb,YAAy,C;MACI,2B;MAAhB,OAAGB,cAAhB



,C;QAAGB,yB;QACZ,oBAAO,cAAP,EAAO,sBAAP,WAAkB,OAAIB,C;;MACJ,OAAO,M;K;IAGX,kC;MAQoB,UACL,M;MAHX,aAAa,eAAU,cAAV,C;MACb,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,oBAAO,cAAP,EAAO,sBAAP,WAAkB,OAAIB,C;;MACJ,OAAO,M;K;IAGX,mC;MAQoB,UACL,M;MAHX,aAAa,gBAAW,cAAX,C;MACb,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,oBAAO,cAAP,EAAO,sBAAP,WAAkB,OAAIB,C;;MACJ,OAAO,M;K;IAGX,oC;MAQoB,UACL,M;MAHX,aAAa,iBAAy,cAAZ,C;MACb,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,oBAAO,cAAP,EAAO,sBAAP,WAAkB,OAAIB,C;;MACJ,OAAO,M;K;IAGX,2B;MAQoB,Q;MADhB,UAAgB,W;MACA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,MrCAiD,SqCAjD,GrCA2D,KAAK,GqCAzD,OrCAoE,KAAx,IAAf,C;;MqCErD,OA AO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAiB,2B;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,M pBXmD,UoBWNd,GpBX8D,KAAK,KoBW5D,OpBXuE,KAAx,CAAhB,C;;MoBavD,OAAO,G;K;IAGX,2B;MA QoB,Q;MADhB,UAAgB,W;MACA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,MrC5BiD,SqC4BjD,GrC5 B2D,KAAK,GAAW,CD2O5C,SsC/MxB,OtC+MkC,KAAAL,GAAiB,GAAtB,CC3O4C,MAAX,IAAf,C;;MqC8BrD, OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAgB,W;MACA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ ,MrC1CiD,SqC0CjD,GrC1C2D,KAAK,GAAW,CC4O5C,SoCIMxB,OpCkMkC,KAAAL,GAAiB,KAAtB,CD5O4C, MAAX,IAAf,C;;MqC4CrD,OAAO,G;K;IC3GX,wB;MAMI,OtCuCkE,YsCvCvD,CtCuCwE,KAAjB,EsCvCID,CtC uC+E,KAA7B,CsCvCvD,KAAJ,GAAy,CAAZ,GAAmB,C;K;IAG9B,wB;MAMI,OrBsCmE,aqBtCxD,CrBsC0E,K AAIB,EqBtCnD,CrBsCiF,KAA9B,CqBtCxD,KAAJ,GAAy,CAAZ,GAAmB,C;K;IAG9B,wB;MAMI,OvCKgF,0Bu CLrE,CvCgP2B,KAAAL,GAAiB,GA3O8B,EuCLhE,CvCgPsB,KAAAL,GAAiB,GA3O8B,CuCLrE,KAAJ,GAAy,CA AZ,GAAmB,C;K;IAG9B,wB;MAMI,OrClIF,0BqCjE,CrCwO2B,KAAAL,GAAiB,KApO+B,EqCjJE,CrCwOsB,KA AL,GAAiB,KApO+B,CqCjE,KAAJ,GAAy,CAAZ,GAAmB,C;K;mFAG9B,yB;MAAA,8C;MAAA,0B;QAOI,OA AO,MAAM,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;MAAA,0B ;QAOI,OAAO,MAAM,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C; MAAA,0B;QAOI,OAAO,MAAM,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB; MAIAUA,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GA AN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAA V,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;MAOc,Q;MADV ,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAAX,C;;M ACvB,OAAO,G;K;IAGX,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO, MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,wB;MAMI,OtCjFkE,YsCiFvD,CtCjFwE,K AAjB,EsCiFID,CtCjF+E,KAA7B,CsCiFvD,KAAJ,GAAy,CAAZ,GAAmB,C;K;IAG9B,wB;MAMI,OrBIFmE,aqB kFxD,CrBIF0E,KAAIB,EqBkFnD,CrBIFiF,KAA9B,CqBkFxD,KAAJ,GAAy,CAAZ,GAAmB,C;K;IAG9B,wB;MA MI,OvCnHgF,0BuCmHrE,CvCwH2B,KAAAL,GAAiB,GA3O8B,EuCmHhE,CvCwHsB,KAAAL,GAAiB,GA3O8B,C uCmHrE,KAAJ,GAAy,CAAZ,GAAmB,C;K;IAG9B,wB;MAMI,OrCpHiF,0BqCoHtE,CrCgH2B,KAAAL,GAAiB,K ApO+B,EqCoHjE,CrCgHsB,KAAAL,GAAiB,KApO+B,CqCoHtE,KAAJ,GAAy,CAAZ,GAAmB,C;K;mFAG9B,yB ;MAAA,8C;MAAA,0B;QAOI,OAAO,MAAM,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C; mFAUA,yB;MAAA,8C;MAAA,0B;QAOI,OAAO,MAAM,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C; O;KAPX,C;mFAUA,yB;MAAA,8C;MAAA,0B;QAOI,OAAO,MAAM,CAAN,EAAS,MAAM,CAAN,EAAS,CAA T,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;MAAA,0B;QAOI,OAAO,MAAM,CAAN,EAAS,MAAM,CAAN, EAAS,CAAT,CAAT,C;O;KAPX,C;IAUA,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QA AU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;MAOc,Q;MADV,UAA U,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,O AAO,G;K;IAGX,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM, SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV, OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IC7OX,6B;M AOI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAc,SAAd,eAAvB,C;MACV,OAAO,SAAK,M;K;IAGhB,6B;M AOI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAc,SAAd,eAAvB,C;MACV,OAAO,SAAK,M;K;IAGhB,mC;M AKI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,SAAK,M;K;IAGzC,mC;MAKI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,

SAAK,M;K;IAGzC,4B;MASI,IAAI,mBAAJ,C;QACI,MAAM,2BAAUb,iBAAc,SAAd,eAAvB,C;MACV,OAAO,S  
AAK,K;K;IAGhB,4B;MASI,IAAI,mBAAJ,C;QACI,MAAM,2BAAUb,iBAAc,SAAd,eAAvB,C;MACV,OAAO,SA  
AK,K;K;IAGhB,kC;MAOI,OAAW,mBAAJ,GAAe,IAAf,GAAYb,SAAK,K;K;IAGzC,kC;MAOI,OAAW,mBAAJ,  
GAAe,IAAf,GAAYb,SAAK,K;K;gFAGzC,yB;MAAA,mC;MAAA,2C;MAAA,4B;QASI,OAAO,kBAAO,cAAP,C;  
O;KATX,C;gFAYA,yB;MAAA,mC;MAAA,2C;MAAA,4B;QASI,OAAO,kBAAO,cAAP,C;O;KATX,C;IAYA,sC;;  
QASQ,OAAc,WAAP,MAAO,EAAS,SAAT,C;;QACHb,+C;UACE,MAAM,2BAAUb,CAAE,QAAzB,C;;UAHV,O;  
;K;IAOJ,sC;;QASQ,OAAc,YAAP,MAAO,EAAU,SAAV,C;;QACHb,+C;UACE,MAAM,2BAAUb,CAAE,QAAzB,  
C;;UAHV,O;;K;4FAOJ,yB;MAAA,mC;MAAA,uD;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;4FAU  
A,yB;MAAA,mC;MAAA,uD;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;IAUA,4C;MAMI,IAAI,mBA  
AJ,C;QACI,OAAO,I;MACX,OAAc,WAAP,MAAO,EAAS,SAAT,C;K;IAGIB,4C;MAMI,IAAI,mBAAJ,C;QACI,O  
AAO,I;MACX,OAAc,YAAP,MAAO,EAAU,SAAV,C;K;oFAGIB,8B;MASI,OAAO,WAAW,IAAX,IAAmB,2BA  
AS,OAAT,C;K;oFAG9B,8B;MASI,OAAO,WAAW,IAAX,IAAmB,2BAAS,OAAT,C;K;IAG9B,uC;MAMI,OAAO  
,2BxCof4B,SwCpFnB,KxCof6B,KAAL,GAAiB,GAAtB,CwCpF5B,C;K;IAGX,uC;MAMI,OAAO,2BxCqF8B,UA  
AW,oBwCrFhC,KxCqF2B,KAAK,CAAL,UAAW,CwCrF9B,C;K;IAGX,uC;MAMI,OAAO,2BxCgG8B,UAAW,oB  
uChGhC,KvCgG2B,KAAK,CAAL,iBAAN,CuChG9B,C;K;IAGX,uC;MAMY,Q;MAAD,cAAC,OtBH4C,UsBG5C,  
KtBhKd,YsBGxG,CtBhwC,CAAN,CsBG7C,wBAA8B,2BAA9B,Q;MAAA,W;QAAqC,oCvC4JR,SuC5JiB,KtBq  
FIB,KjBuEW,QAAV,CuC5JQ,C;;MAA5C,a;K;IAGJ,uC;MAMI,OAAO,2BtCiD4B,SsCjDnB,KtCiD6B,KAAL,GA  
AiB,KAAtB,CsCjD5B,C;K;IAGX,uC;MAMI,OAAO,2BtCkD8B,UAAW,oBsCIDhC,KtCkD2B,KAAK,CAAL,YA  
AN,CsCID9B,C;K;IAGX,kC;MASI,OAAO,uCAAgB,yBxC2BY,SwC3BI,SxC2BM,KAAL,GAAiB,GAAtB,CwC3  
BZ,ExC2BY,SwC3BmB,ExC2BT,KAAL,GAAiB,GAAtB,CwC3BZ,EAA4C,EAA5C,C;K;IAG3B,kC;MASI,OAA  
O,uCAAgB,yBAAGB,SAAhB,EAAsB,EAAtB,EAA0B,EAA1B,C;K;IAG3B,kC;MASI,OAAO,wCAAiB,yBAAGB,  
SAAhB,EAAsB,EAAtB,M;K;IAG5B,kC;MASI,OAAO,uCAAgB,yBtCRY,SsCQI,StCRM,KAAL,GAAiB,KAAtB,  
CsCQZ,EtCRY,SsCQmB,EtCRT,KAAL,GAAiB,KAAtB,CsCQZ,EAA4C,EAA5C,C;K;wFAG3B,yB;MAAA,yC;M  
AAA,gC;QASI,OAAO,iBAAM,EAAN,C;O;KATX,C;wFAYA,yB;MAAA,yC;MAAA,gC;QASI,OAAO,iBAAM,E  
AAN,C;O;KATX,C;wFAYA,yB;MAAA,yC;MAAA,gC;QASI,OAAO,iBAAM,EAAN,C;O;KATX,C;wFAYA,yB;  
MAAA,yC;MAAA,gC;QASI,OAAO,iBAAM,EAAN,C;O;KATX,C;IAYA,gC;MAMI,OAAO,uCAAgB,yBAAGB,c  
AAhB,EAAsB,eAAtB,EAA6B,CAAC,cAAD,IAA7B,C;K;IAG3B,gC;MAMI,OAAO,wCAAiB,yBAAGB,cAAhB,E  
AAsB,eAAtB,EAA8B,cAAD,aAA7B,C;K;IAG5B,iC;MAMI,oBAAoB,OAAO,CAA3B,EAA8B,IAA9B,C;MACA,  
OAAO,uCAAgB,yBAAGB,eAAhB,EAAuB,cAAvB,EAAiC,SAAK,KAAL,GAAY,CAAhB,GAAMB,IAANB,GAA  
6B,CAAC,IAAD,IAA1D,C;K;IAG3B,iC;MAMI,oBAAoB,kBAAO,CAA3B,EAA8B,IAA9B,C;MACA,OAAO,wC  
AAiB,yBAAGB,eAAhB,EAAuB,cAAvB,EAAiC,SAAK,KAAL,cAAY,CAAhB,GAAMB,IAANB,GAA8B,IAAD,a  
AA1D,C;K;IAG5B,iC;MAQI,IxCvXgF,OBwCuX5E,ExC5IkC,KAAL,GAAiB,GA3O8B,EwCuXtE,6BAAM,UxC5I  
sB,KAAL,GAAiB,GA3O8B,CwCuX5E,KAAJ,C;QAA2B,OAAO,iCAAU,M;MACHC,WxC3GuB,SwC2G5B,SxC3  
GsC,KAAL,GAAiB,GAAtB,C;MwC2GV,YAAK,W;MAA9B,OvCzL6D,oBAhJP,SAAU,CD8N7B,SwC2GV,ExC3  
GoB,KAAL,GAAiB,GAAtB,CC9N6B,MAAK,GDAK,KCAO,KAAZ,IAAf,CAgJO,C;K;IuC4LjE,iC;MAQI,IvCnX  
kE,YuCmX9D,EvCnX+E,KAAjB,EuCmXxD,4BAAK,UvCnXgF,KAA7B,CuCmX9D,KAAJ,C;QAA0B,OAAO,iC  
AAU,M;MAC3C,OvCrM6D,cuCqMtD,SvCrMsD,EAhJP,SuCqVtC,EvCrVgD,KAAK,GAAY,CuCqV5D,WvCrV4  
D,MAAZ,IAAf,CAgJO,C;K;IuCwMjE,iC;MAQI,ItBvXmE,asBuX/D,EtBvXiF,KAAIB,EsBuXzD,6BAAM,UtBvXi  
F,KAA9B,CsBuX/D,KAAJ,C;QAA2B,OAAO,kCAAW,M;MAC7C,OtBjN+D,iBsBiNxD,StBjNwD,EA7IP,UsB8V  
xC,EtB9VmD,KAAK,UAAZ,CjByP/C,UAAW,oBAAL,CuCqGtB,WvCrGsB,MAAK,CAAL,iBAAN,CiBzP+C,M  
AAZ,CAAhB,CA6IO,C;K;IsBoNnE,iC;MAQI,ItCnZiF,0BsCmZ7E,EtC/KkC,KAAL,GAAiB,KApO+B,EsCmZvE,  
8BAAO,UtC/KqB,KAAL,GAAiB,KApO+B,CsCmZ7E,KAAJ,C;QAA4B,OAAO,iCAAU,M;MACjC,WtC9IuB,Ss  
C8I5B,StC9IsC,KAAL,GAAiB,KAAtB,C;MsC8IV,YAAK,W;MAA9B,OvC7N6D,oBAhJP,SAAU,CC+N7B,SsC8I  
V,EtC9IoB,KAAL,GAAiB,KAAtB,CD/N6B,MAAK,GCAK,KDAO,KAAZ,IAAf,CAgJO,C;K;IuCgOjE,kD;MAUI,  
OvCzZkE,YuCyZvD,SvCzZwE,KAAjB,EuCyZhD,YvCzZ6E,KAA7B,CuCyZvD,IAAJ,GAAYB,YAAzB,GAA2C,  
S;K;IAGtD,kD;MAUI,OtB9ZmE,asB8ZxD,StB9Z0E,KAAIB,EsB8ZjD,YtB9Z+E,KAA9B,CsB8ZxD,IAAJ,GAAY  
B,YAAzB,GAA2C,S;K;IAGtD,kD;MAUI,OxCncgF,OBwCmcrE,SxCxN2B,KAAL,GAAiB,GA3O8B,EwCmc9D,Y  
xCxNoB,KAAL,GAAiB,GA3O8B,CwCmcrE,IAAJ,GAAYB,YAAzB,GAA2C,S;K;IAGtD,kD;MAUI,OtCxcif,OBs

CwctE,StCpO2B,KAAL,GAAiB,KApO+B,EsCwc/D,YtCpOoB,KAAL,GAAiB,KApO+B,CsCwctE,IAAJ,GAAyB, YAAzB,GAA2C,S;K;IAGtD,iD;MAUI,OvC7ckE,YuC6cvD,SvC7cwE,KAAjB,EUc6chD,YvC7c6E,KAA7B,CuC6 cvD,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAUI,OtBldmE,asBkdxD,StBld0E,KAAiB,EsBkdjD,YtBld+E ,KAA9B,CsBkdxD,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAUI,OxCvfgF,0BwCufR,E,SxC5Q2B,KAAL, GAAiB,GA3O8B,EwCuf9D,YxC5QoB,KAAL,GAAiB,GA3O8B,CwCufR,E,IAAJ,GAAyB,YAAzB,GAA2C,S;K;I AGtD,iD;MAUI,OtC5fiF,0BsC4fiE,StCXR2B,KAAL,GAAiB,KApO+B,EsC4f/D,YtCxRoB,KAAL,GAAiB,KApO +B,CsC4fiE,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,4D;MAUI,IvCjgBkE,YuCigB9D,YvCjgB+E,KAAjB,EUc igB/C,YvCjgB4E,KAA7B,CuCigB9D,IAAJ,C;QAAiC,MAAM,gCAAyB,oDAaiD,YAAjD,8BAAoF,YAApF,MA AzB,C;MACvC,IvClgBkE,YuCkgB9D,SvClgB+E,KAAjB,EUcKgBvD,YvClgBoF,KAA7B,CuCkgB9D,IAAJ,C;Q AAyB,OAAO,Y;MACHC,IvCngBkE,YuCmgB9D,SvCngB+E,KAAjB,EUcMgBvD,YvCngBoF,KAA7B,CuCmgB9 D,IAAJ,C;QAAyB,OAAO,Y;MACHC,OAAO,S;K;IAGX,4D;MAUI,ItBzgBmE,asBygB/D,YtBzgBiF,KAAiB,EsBy gBhD,YtBzgB8E,KAA9B,CsBygB/D,IAAJ,C;QAAiC,MAAM,gCAAyB,oDAaiD,YAAjD,8BAAoF,YAApF,MAA zB,C;MACvC,ItB1gBmE,asB0gB/D,StB1gBiF,KAAiB,EsB0gBxD,YtB1gBsF,KAA9B,CsB0gB/D,IAAJ,C;QAAy B,OAAO,Y;MACHC,ItB3gBmE,asB2gB/D,StB3gBiF,KAAiB,EsB2gBxD,YtB3gBsF,KAA9B,CsB2gB/D,IAAJ,C; QAAyB,OAAO,Y;MACHC,OAAO,S;K;IAGX,4D;MAUI,IxCjjBgF,0BwCijB5E,YxCtUkC,KAAL,GAAiB,GA3O8 B,EwCijB7D,YxCtUmB,KAAL,GAAiB,GA3O8B,CwCijB5E,IAAJ,C;QAAiC,MAAM,gCAAyB,oDAaiD,YAAjD ,8BAAoF,YAApF,MAAzB,C;MACvC,IxCljBgF,0BwCkjB5E,SxCvUkC,KAAL,GAAiB,GA3O8B,EwCkjBrE,YxC vU2B,KAAL,GAAiB,GA3O8B,CwCkjB5E,IAAJ,C;QAAyB,OAAO,Y;MACHC,IxCnjBgF,0BwCmjB5E,SxCxUkC ,KAAL,GAAiB,GA3O8B,EwCmjBrE,YxCxU2B,KAAL,GAAiB,GA3O8B,CwCmjB5E,IAAJ,C;QAAyB,OAAO,Y ;MACHC,OAAO,S;K;IAGX,4D;MAUI,ItCzjBiF,0BsCyjB7E,YtCrVkc,KAAL,GAAiB,KApO+B,EsCyjB9D,YtCr VmB,KAAL,GAAiB,KApO+B,CsCyjB7E,IAAJ,C;QAAiC,MAAM,gCAAyB,oDAaiD,YAAjD,8BAAoF,YAApF, MAAzB,C;MACvC,ItC1jBiF,0BsC0jB7E,StCtVkc,KAAL,GAAiB,KApO+B,EsC0jBtE,YtCtV2B,KAAL,GAAiB, KApO+B,CsC0jB7E,IAAJ,C;QAAyB,OAAO,Y;MACHC,ItC3jBiF,0BsC2jB7E,StCvVkc,KAAL,GAAiB,KApO+B ,EsC2jBtE,YtCvV2B,KAAL,GAAiB,KApO+B,CsC2jB7E,IAAJ,C;QAAyB,OAAO,Y;MACHC,OAAO,S;K;IAGX, uC;MAcW,Q;MAJP,IAAI,8CAAJ,C;QACI,OAAY,WAAL,SAAK,EAAe,KAAf,C;;MAEhB,IAAI,KAAM,UAAV, C;QAAqB,MAAM,gCAAyB,4CAAYC,KAAzC,MAAzB,C;MAEvB,IvCtkB8D,YuCskB9D,SvCtkB+E,KAAjB,EU cSkBvD,KAAM,MvCtkB8E,KAA7B,CuCskB9D,K;QAA4B,OAAN,KAAM,M;;QAC5B,IvCvkB8D,YuCukB9D,S vCvkB+E,KAAjB,EUcukBvD,KAAM,avCvkB8E,KAA7B,CuCukB9D,K;UAAmC,OAAN,KAAM,a;;UAC3B,gB;; MAHZ,W;K;IAOJ,uC;MAcW,Q;MAJP,IAAI,8CAAJ,C;QACI,OAAY,WAAL,SAAK,EAAgB,KAAhB,C;;MAEhB ,IAAI,KAAM,UAAV,C;QAAqB,MAAM,gCAAyB,4CAAYC,KAAzC,MAAzB,C;MAEvB,ItBnlB+D,asBmlB/D,St BnlBiF,KAAiB,EsBmlBxD,KAAM,MtBnlBgF,KAA9B,CsBmlB/D,K;QAA4B,OAAN,KAAM,M;;QAC5B,ItBplB +D,asBolB/D,StBplBiF,KAAiB,EsBolBxD,KAAM,atBplBgF,KAA9B,CsBolB/D,K;UAAmC,OAAN,KAAM,a;;U AC3B,gB;;MAHZ,W;K;ICvoBJ,2B;MAUoB,Q;MADhB,UAAgB,W;MACA,2B;MAAhB,OAAGB,cAAhB,C;QAA gB,yB;QACZ,MxCoDiD,SwCpDjD,GxCoD2D,KAAK,GwCpDzD,OxCoDoE,KAAAX,IAAf,C;;MwClDrD,OAAO, G;K;IAGX,2B;MAUoB,Q;MADhB,UAAiB,2B;MACD,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QACZ,MvBu CmD,UuBvCnD,GvBuC8D,KAAK,KuBvC5D,OvBuCuE,KAAAX,CAAhB,C;;MuBrCvD,OAAO,G;K;IAGX,2B;M AUoB,Q;MADhB,UAAgB,W;MACA,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QACZ,MxCoBiD,SwCpBjD,Gx CoB2D,KAAK,GAAW,CD2O5C,SyC/PxB,OzC+PkC,KAAL,GAAiB,GAAtB,CC3O4C,MAAX,IAAf,C;;MwClBr D,OAAO,G;K;IAGX,2B;MAUoB,Q;MADhB,UAAgB,W;MACA,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QA CZ,MxClId,SwCjJd,GxCi2D,KAAK,GAAW,CC4O5C,SuChPxB,OvCgPkC,KAAL,GAAiB,KAAtB,CD5O4C,M AAX,IAAf,C;;MwCFrD,OAAO,G;K;,,,,;ICuCP,iD;MAAA,qE;MAAGB,4B;MANpB,uC;MAMI,Y;K;IACA,4D;MA AA,qE;MAAGC,wBAAM,OAAN,Q;MAPpC,uC;MAOI,Y;K;IACA,mE;MAAA,qE;MAAmD,6BAAM,OAAN,EA Ae,KAAf,C;MARvD,uC;MAQI,Y;K;IACA,0D;MAAA,qE;MAAiC,wBAAM,KAAN,Q;MATrC,uC;MASI,Y;K;IC xGJ,gC;K;,,,,;ICuBoC,wC;8BAAsC,O;K;,,,,,;yCC0rTE,6B;MASI,MAAM,yB;K;,,,,,;0CAyDV,s B;MASI,OAAO,I;K;,,,,,;IC1Xf,gB;MAAA,oB;K;8BAII,Y;MAA0B,oB;K;,,;IAJ9B,4B;MAAA,2B;QAAA, U;MAAA,oB;K;ICEA,yC;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,uC;MAAA,0C;O;MAII,kE;MAEA,wF;MAE A,oF;MAEA,wE;MAEA,kE;MAEA,oF;MAEA,sF;MAEA,8E;MAEA,wE;MAEA,sF;MAEA,uF;MAEA,iE;MAEA, 6E;MAEA,iE;MAEA,2E;K;;IA5BA,8C;MAAA,6B;MAAA,sC;K;;IAEA,yD;MAAA,6B;MAAA,iD;K;;IAEA,uD;M

AAA,6B;MAAA,+C;K;;IAEA,iD;MAAA,6B;MAAA,yC;K;;IAEA,8C;MAAA,6B;MAAA,sC;K;;IAEA,uD;MAAA,6B;MAAA,+C;K;;IAEA,wD;MAAA,6B;MAAA,gD;K;;IAEA,oD;MAAA,6B;MAAA,4C;K;;IAEA,iD;MAAA,6B;MAAA,yC;K;;IAEA,wD;MAAA,6B;MAAA,gD;K;;IAEA,wD;MAAA,6B;MAAA,gD;K;;IAEA,6C;MAAA,6B;MAAA,qC;K;;IAEA,mD;MAAA,6B;MAAA,2C;K;;IAEA,6C;MAAA,6B;MAAA,qC;K;;IAEA,kD;MAAA,6B;MAAA,0C;K;;IAhCJ,mC;MAAA,+oB;K;;IAAA,wC;MAAA,a;AAAA,O;UAAA,2C;aAAA,kB;UAAA,sD;aAAA,gB;UAAA,oD;aAAA,U;UAAA,8C;aAAA,O;UAAA,2C;aAAA,gB;UAAA,oD;aAAA,iB;UAAA,qD;aAAA,a;UAAA,iD;aAAA,U;UAAA,8C;aAAA,iB;UAAA,qD;aAAA,iB;UAAA,qD;aAAA,M;UAAA,0C;aAAA,Y;UAAA,gD;aAAA,M;UAAA,0C;aAAA,W;UAAA,+C;;UAAA,uE;;K;;IAqCA,4C;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,0C;MAAA,6C;O;MAMI,0E;MAEA,0E;MAEA,4E;K;;IAJA,kD;MAAA,gC;MAAA,0C;K;;IAEA,kD;MAAA,gC;MAAA,0C;K;;IAEA,mD;MAAA,gC;MAAA,2C;K;;IAVJ,sC;MAAA,sI;K;;IAAA,2C;MAAA,a;AAAA,Q;UAAA,+C;aAAA,Q;UAAA,+C;aAAA,S;UAAA,gD;;UAAA,0E;;K;;IAwB8B,gC;MAAC,oC;K;;IAQE,0B;MAAC,qB;QAAA,iD;MAAA,kB;K;;IAEIC,sB;K;;IAMA,4B;K;;ICxFQ,kD;MAAA,8B;MACI,aAAy,C;K;oDACZ,Y;MAAyB,oBAAQ,gBAAI,O;K;iDACrC,Y;MAAgD,Q;MAA1B,IAAI,aAAQ,gBAAI,OAaHb,C;QAAA,OAAsB,iBAAI,iBAAJ,EAAL,yBAAJ,O;;QAAkB,MAAM,2BAAYB,UAAF,WAAvB,C;K;;IAPhF,oC;MAEL,IAD8D,IAC9D,S;QACI,UAA0B,K;QAF0B,2C;;QAAA,QAAM,IAAN,C;eASxD,c;YATwD,OAStC,qBAAqB,KAARb,C;eACIB,W;YAVwD,OAuZC,kBAAkB,KAAIB,C;eACf,Y;YAXwD,OAWxC,mBAAMB,KAAAnB,C;eAChB,W;YAZwD,OAYzC,kBAAkB,KAAIB,C;eACf,U;YAbwD,OAa1C,iBAAiB,KAAjB,C;eACd,W;YAdwD,OaczC,kBAAkB,KAAIB,C;eACf,Y;YafwD,OAexC,mBAAMB,KAAAnB,C;eAChB,a;YAhBwD,OAgBvC,oBAAoB,KAApB,C;;YACT,MAAM,6BAAsB,2DAA+C,IAA/C,CAAtB,C;;K;IAIuC,2D;MAAA,kC;MAAS,0B;MAC9D,aAAy,C;K;2DACZ,Y;MAAyB,oBAAQ,kBAAM,O;K;;+DACvC,Y;MAA2D,Q;MAA9B,IAAI,aAAQ,kBAAM,OAaIB,C;QAAA,OAaWb,mBAAM,iBAAN,EAAM,yBAAN,O;;QAAoB,MAAM,2BAAYB,UAAF,WAAvB,C;K;;IAJnF,qC;MACyD,oD;K;IAON,wD;MAAA,kC;MAAS,uB;MACxD,aAAy,C;K;wDACZ,Y;MAAyB,oBAAQ,kBAAM,O;K;yDACvC,Y;MAAwD,Q;MAA9B,IAAI,aAAQ,kBAAM,OAaIB,C;QAAA,OAaWb,mBAAM,iBAAN,EAAM,yBAAN,O;;QAAoB,MAAM,2BAAYB,UAAF,WAAvB,C;K;;IAJhF,kC;MACmD,iD;K;IAOE,yD;MAAA,kC;MAAS,wB;MAC1D,aAAy,C;K;yDACZ,Y;MAAyB,oBAAQ,kBAAM,O;K;2DACvC,Y;MAAyD,Q;MAA9B,IAAI,aAAQ,kBAAM,OAaIB,C;QAAA,OAaWb,mBAAM,iBAAN,EAAM,yBAAN,O;;QAAoB,MAAM,2BAAYB,UAAF,WAAvB,C;K;;IAJf,mC;MACqD,kD;K;IAOF,wD;MAAA,kC;MAAS,uB;MACxD,aAAy,C;K;wDACZ,Y;MAAyB,oBAAQ,kBAAM,O;K;yDACvC,Y;MAAwD,Q;MAA9B,IAAI,aAAQ,kBAAM,OAaIB,C;QAAA,OAaWb,mBAAM,iBAAN,EAAM,yBAAN,O;;QAAoB,MAAM,2BAAYB,UAAF,WAAvB,C;K;;IAJhF,kC;MACmD,iD;K;IAOF,uD;MAAA,kC;MAAS,sB;MACtD,aAAy,C;K;uDACZ,Y;MAAyB,oBAAQ,kBAAM,O;K;uDACvC,Y;MAAuD,Q;MAA9B,IAAI,aAAQ,kBAAM,OAaIB,C;QAAA,OAaWb,mBAAM,iBAAN,EAAM,yBAAN,O;;QAAoB,MAAM,2BAAYB,UAAF,WAAvB,C;K;;IAJ/E,iC;MACiD,gD;K;IAOI,yD;MAAA,kC;MAAS,wB;MAC1D,aAAy,C;K;yDACZ,Y;MAAyB,oBAAQ,kBAAM,O;K;2DACvC,Y;MAAyD,Q;MAA9B,IAAI,aAAQ,kBAAM,OAaIB,C;QAAA,OAaWb,mBAAM,iBAAN,EAAM,yBAAN,O;;QAAoB,MAAM,2BAAYB,UAAF,WAAvB,C;K;;IAJf,mC;MACqD,kD;K;IAOE,0D;MAAA,kC;MAAS,yB;MAC5D,aAAy,C;K;0DACZ,Y;MAAyB,oBAAQ,kBAAM,O;K;6DACvC,Y;MAA0D,Q;MAA9B,IAAI,aAAQ,kBAAM,OAaIB,C;QAAA,OAaWb,mBAAM,iBAAN,EAAM,yBAAN,O;;QAAoB,MAAM,2BAAYB,UAAF,WAAvB,C;K;;IAJIF,oC;MACuD,mD;K;IAOJ,wD;MAAA,kC;MAAS,uB;MACxD,aAAy,C;K;wDACZ,Y;MAAyB,oBAAQ,kBAAM,O;K;yDACvC,Y;MAAwD,Q;MAA9B,IAAI,aAAQ,kBAAM,OAaIB,C;QAAA,OAaWb,mBAAM,iBAAN,EAAM,yBAAN,O;;QAAoB,MAAM,2BAAYB,UAAF,WAAvB,C;K;;IAJhF,kC;MACmD,iD;K;IAOpB,gC;MAAC,wB;K;;IAEHc,+B;MAC8C,MAAM,mC;K;IAEpD,8C;MAEL,IAAI,qBAAJ,C;QACI,OAaO,CtBkKiF,WsBIKrE,UtBkKqE,EsBIKzD,QtBkKyD,C;;QsBhKxF,OAAS,CAAY,qBAAsB,UAAtB,EAaKc,QAaIC,C;;K;IAI7B,2C;MAEL,IAAI,KAAY,kBAAhB,C;QAGI,KAAY,mBAAkB,QAaIB,C;;QAEH,QAAT,SAA+C,CAaIB,IAAjC,KAAiC,EAaKb,O;;K;IAIvD,sC;MAGwB,Q;MADpB,gBAAGb,IAAhB,KAAgB,E;MACI,IAAI,OCnGkB,ODmGT,OAAT,EAaQb,WAArB,CAAJ,C;QACHb,OAAL,aAAJ,GAAMb,KAAM,WAAzB,GAAyC,I;;QAEzC,c;;MAHJ,wB;MAKA,kBAAkB,K;MACIB,iBAAiB,W;MACjB,OAaO,S;K;IAIa,sB;MAAC,U;K;iCACrB,iB;MACI,OAaO,mCAAsB,WAAK,KAAM,E;K;mCAG5C,Y;MACI,OAaO,M;K;mCAGX,Y;MACI,OAaUc,oBAAnB,UAA5B,IAAe,EAaA,CAAMb,C;K;0CAG3C,iB;MACI,OAAR,IAAI,EAaW,GAAN,K;K;kCAGL,Y;MAEL,OAaO,M;K;;+DAIf,gB;MAEL,YAAy,MAAY,IAAK,OAajB,C;MACZ,sBAAU,IAAV,a;QACI,UAAU,KAAK,CAAL,C;QACV,IAAI,oBAAJ,C;UACI,MAAM,CAAN,I

AAW,EAAS,MAAM,MAAK,GAAL,C;;UAE1B,MAAM,CAAN,IAAW,G;;;MAGnB,OAAO,EAAS,OAAO,OAA  
M,EAAN,EAAGB,KAAhB,C;K;IAG3B,2B;MAMW,WAAO,S;MAIbD,YAAy,MAAY,IAAK,OAAjB,C;MACZ,sB  
AAU,IAAV,a;QACI,UAAU,KAAK,CAAL,C;QACV,IAAI,oBAAJ,C;UACI,MAAM,CAAN,IAAW,EAAS,MAAM  
,MAAK,GAAL,C;;UAE1B,MAAM,CAAN,IAAW,G;;;MAYnB,OATO,EAAS,OAAO,OAAM,EAAN,EAAGB,KA  
AhB,C;K;IAY3B,oC;MAWI,WAAqB,S;MACrB,IAAI,qBAAmB,CAAY,OAAd,KAA2B,SAAhD,C;QAJCA,YAA  
Y,MAKCM,IAICW,OAAjB,C;QACZ,sBAiCkB,IAjClB,a;UACI,UAGCc,IAhCJ,CAAK,CAAL,C;UACV,IAAI,oBA  
AJ,C;YACI,MAAM,CAAN,IAAW,EAAS,MAAM,MAAK,GAAL,C;;YAE1B,MAAM,CAAN,IAAW,G;;;QA4Bf,  
OAzBG,EAAS,OAAO,OAAM,EAAN,EAAGB,KAAhB,C;;QA2BnB,WAAW,C;QACX,0BAAU,IAAV,e;UACY,I  
AAoB,I;UAA5B,eAAQ,QAAoB,OAAPB,IAAQ,CAAH,GAAG,CAAY,OAAPB,oCAAR,K;;QAEJ,aAAa,IAAJB,C  
AAC,YAAGB,CAAH,IAAG,C;QE3FjB,IF4FyB,CE5FhB,OAAL,KAAkB,SAAtB,C;UF4F4B,ME3FxB,UF2FqB,C  
E3FF,O;;QF4FnB,OAAO,C;QACP,0BAAU,IAAV,e;UAE0B,YACX,M;UAFX,YAAU,IAAQ,CAAH,GAAG,C;UA  
CI,SAAJ,KAAI,O;UAAtB,aAAU,CAAV,kB;YACI,OAAO,aAAP,EAAO,qBAAP,YAAiB,MAAI,CAAJ,C;;QAGz  
B,OAAO,M;;K;IAIf,0B;MACgC,WAAS,c;MAAT,YAAhC,EAAE,MAAM,KAAiD,CAA3C,SAA2C,C;MAWrD,e  
AAiB,I;MAXW,OAYrB,K;K;IAVX,uB;MAC6B,WAAS,W;MAAT,YAAsB,IAA/C,WAA+C,CAAnC,SAAmC,C;  
MAQ/C,eAAiB,I;MARQ,OASIB,K;K;IAPX,uB;MAC6B,WAAS,W;MAAT,YAA7B,EAAE,MAAM,KAA2C,CAA  
rC,SAAqC,C;MAK/C,eAAiB,I;MALQ,OAMIB,K;K;2DAJX,uB;MAGI,eAAiB,I;MACjB,OAAO,K;K;kEG9MX,y  
B;MAAA,0B;MAAA,uB;QASI,OAAoB,OAAb,IjD0Q+B,KAAI,GAaiB,KiD1Q9B,C;O;KATxB,C;IClqC,2C;MA  
AC,8C;MACiC,eAAsB,C;MACTB,wBAA+B,C;MAC/B,gBAA6B,I;MAC7B,mBAA5C,I;MACTC,qBAAyC,I;MAE  
zC,yBAAgD,yBAAmB,Q;MAEnE,sBAAgD,I;K;wFAFhD,Y;MAAA,6B;K;0CAIA,Y;MAEY,kBADR,M;MAAA,  
U;MAAA,2C;QAAA,e;;QAES,gBADD,2CAAQ,yCAAR,gDAaWd,IAAxD,6BAAiE,I;QACzD,sBpCwEd,S;QoC1  
EF,SpC2EG,S;;MoC3EH,a;K;iDAIJ,kB;MACI,kBAAC,IAAd,C;MACiC,oB;MCuBrB,Q;MADR,IdtBsB,MCsBtB,  
W;QADJ,mBACiB,I;;QADjB,mBAEY,QDvBc,MCuBd,+D;;MDvBZ,yC;MACA,2BAAmC,MAAO,kBAA1C,C;M  
AGA,OAAO,IAAP,C;QpCoCY,gBoCnCH,S;;QACD,iBAAiB,8B;QAGjB,IAAI,0BAAJ,C;UACI,qBAAC,e;;UAEd,  
oBAAQ,0B;UACR,wBAAy,kB;;UAIZ,cAAc,oB;UACd,IAAI,YAAy,yBAAhB,C;YAAqC,M;UACrC,kBAAgB,O  
;UACHB,qBAAmB,I;;UAEnB,kBAAgB,I;UACHB,qBAAmB,S;;QAGvB,gC;QAEA,IAAI,wCAAJ,C;UAEI,YAAU,  
U;;UAGV,U;UAAA,0C;YETHB,8BDgDQ,WAAO,qBAAP,CChDR,C;YFSgB,a;;YAAA,a;UAAA,mB;YAEK,UEp  
BrB,oBDgDQ,WD5B+B,eC4B/B,CChDR,C;;UFqBgB,M;;K;mDAMhB,Y;MACI,kBAAKB,mB;MACIB,IAAI,uB  
AAuB,gBAAgB,IAA3C,C;QACI,uCAAQ,yCAAR,EAAmC,wCAA+B,WAA/B,C;;MAEvC,sBAAoB,mC;K;;IAM5  
B,iC;MAAA,qC;K;gGAEQ,Y;MvC0DyC,MAAM,6BuC1DjC,uCvC0D+D,WAA9B,C;K;yDuCxDnD,kB;MvCwD6  
C,MAAM,6BuCvDzC,uCvCuDuE,WAA9B,C;K;+CuCpDnD,Y;MAAkC,8C;K;;IARtC,6C;MAAA,4C;QAAA,2B;;  
MAAA,qC;K;IGyDA,mG;IAAA,yH;IAAA,6F;MAKW,kC;MAAS,4C;K;IALpB,sEAMQ,Y;MACI,Q;MAAA,sC;Q  
AAiB,U;;MACjB,OAAO,oB;K;IARnB,6G;sJAJIA,iC;MAGBU,OAAK,SAAL,CAAIb,UAAjB,EAA6B,KAA7B,C;  
K;wJAEV,2C;MAiBU,OAAK,SAAL,CAAIb,QAAjB,EAA2B,UAA3B,EAAuC,KAAvC,C;K;wJAEV,kD;MAKU,  
OAAK,SAAL,CAAIb,QAAjB,EAA2B,KAA3B,EAAkC,UAAiC,EAA8C,KAA9C,C;K;IAGC6C,oG;MAAA,mB;Q  
AC3C,OAAK,iCAAL,CAAIb,kBAAjB,C;O;K;IA/BZ,6D;MA0BI,IAAS,SAAY,OAAjB,IAA2B,CAA/B,C;QAAA,  
OAES,SAAL,CAAIb,UAAjB,EAA6B,IAA7B,C;;QA8D0B,Q;QAhE9B,4DAImD,0DAJnD,EAGe8B,qBA5DS,UA  
4DT,qCAhE9B,C;;K;IAwCmD,wH;MAAA,mB;QAC3C,OAAK,iCAAL,CAAIb,gBAAjB,EAA2B,kBAA3B,C;O;  
K;IAhCZ,yE;MA2BI,IAAS,SAAY,OAAjB,IAA2B,CAA/B,C;QAAA,OAES,SAAL,CAAIb,QAAjB,EAA2B,UAA  
3B,EAAuC,IAAvC,C;;QA0B0B,Q;QA5B9B,4DAImD,sEAJnD,EA4B8B,qBAxBS,UAWBT,qCA5B9B,C;;K;IASJ,  
gC;MAWK,kBAAD,M;MAAA,kBAAC,qEAAD,4DAA2C,S;K;6CAG/C,yB;MAAA,mG;MAAA,yH;MAAA,6F;Q  
AKW,kC;QAAS,4C;O;MALpB,sEAMQ,Y;QACI,Q;QAAA,sC;UAAiB,U;;QACjB,OAAO,oB;O;MARnB,6G;MA  
AA,oC;QAKkC,Q;QAA9B,mEAA8B,oEAA9B,C;O;KALJ,C;IFC7HA,a;MAC6C,OAAA,MAAA,YAAW,CAAX,C;  
K;ICM3B,iC;;MAA6E,Q;MAAA,+BAAS,I;sCAAIB,O,2DAAA,O;;;K;,,,,,;IAC/F,2B;MAAA,iD;MAAuB,oBAA  
K,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,sC;MAAA,iD;MAAuC,oBAAK,OAAL,EAAc,IAAd,C;MAAvC,Y;  
K;IACA,oC;MAAA,iD;MAAwC,oBAAK,SAAL,EAGB,KAAhB,C;MAAxC,Y;K;IAI+B,mC;;MAA6E,Q;MAAA,  
+BAAS,I;sCAAIB,O,2DAAA,O;;;K;,,,,,;IACnG,+B;MAAA,mD;MAAuB,sBAAK,IAAL,EAAW,IAAX,C;MAAv  
B,Y;K;IACA,0C;MAAA,mD;MAAuC,sBAAK,OAAL,EAAc,IAAd,C;MAAvC,Y;K;IACA,wC;MAAA,mD;MAA  
wC,sBAAK,SAAL,EAGB,KAAhB,C;MAAxC,Y;K;IAGsC,0C;MAA0D,qBAAU,OAAY,EAAmB,KAAAnB,C;;K;

IACHg,sC;MAAA,0D;MAAuB,6BAAK,IAAL,EA AW,IAAX,C;MAAvB,Y;K;IACA,iD;MAAA,0D;MAAuC,6BA AK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IACA,+C;MAAA,0D;MAAwC,6BAAK,SAAL,EAAGB,KAAhB,C;MAA xC,Y;K;IAG8C,kD;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IACxG,8C;MAAA,kE;MAAuB,qCAAK,IA AL,EA AW,IAAX,C;MAAvB,Y;K;IACA,yD;MAAA,kE;MAAuC,qCAAK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IA CA,uD;MAAA,kE;MAAwC,qCAAK,SAAL,EAAGB,KAAhB,C;MAAxC,Y;K;IAG2C,+C;MAA0D,4BAAiB,OAAj B,EAA0B,KAA1B,C;;K;;IACrG,2C;MAAA,+D;MAAuB,kCAAK,IAAL,EA AW,IAAX,C;MAAvB,Y;K;IACA,sD; MAAA,+D;MAAuC,kCAAK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IACA,oD;MAAA,+D;MAAwC,kCAAK,SAAL, EAAGB,KAAhB,C;MAAxC,Y;K;IAG+C,4C;8BAAwD,O;;K;;IACvG,+C;MAAA,mE;MAAuB,sCAAK,IAAL,C;M AAvB,Y;K;IAGqD,yD;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IAC/G,qD;MAAA,yE;MAAuB,4CAAK, IAAL,EA AW,IAAX,C;MAAvB,Y;K;IACA,gE;MAAA,yE;MAAuC,4CAAK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;I ACA,8D;MAAA,yE;MAAwC,4CAAK,SAAL,EAAGB,KAAhB,C;MAAxC,Y;K;IAGmD,uD;MAA0D,4BAAiB,O AAjB,EAA0B,KAA1B,C;;K;;IAC7G,mD;MAAA,uE;MAAuB,0CAAK,IAAL,EA AW,IAAX,C;MAAvB,Y;K;IAC A,8D;MAAA,uE;MAAuC,0CAAK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IACA,4D;MAAA,uE;MAAwC,0CAAK,S AAL,EAAGB,KAAhB,C;MAAxC,Y;K;IAI2C,wC;sCAAGe,O;;K;;IAC3G,2C;MAAA,+D;MAAuB,kCAAK,IAAL, C;MAAvB,Y;K;IAI0C,uC;8BAAwD,O;;K;;IAClG,0C;MAAA,8D;MAAuB,iCAAK,IAAL,C;MAAvB,Y;K;IAGwC ,qC;8BAAwD,O;;K;;IACHg,wC;MAAA,4D;MAAuB,+BAAK,IAAL,C;MAAvB,Y;K;IAIJ,wC;MACmD,mBAAM, OAAN,EA Ae,KAAf,C;;K;;IAC/C,oC;MAAA,wD;MAAuB,sBAAK,IAAL,Q;MAAvB,Y;K;IACA,+C;MAAA,wD; MAAGC,2BAAK,OAAL,EAAC,IAAd,C;MAAhC,Y;K;IACA,+C;MAAA,wD;MAAiD,IAAY,I;MAAZB,2BAAa,SA AR,OAAQ,CAAb,EAAYB,sDAAZB,C;MAApC,Y;K;IAG4C,yC;8BAAwD,O;;K;;IACpG,4C;MAAA,gE;MAAuB, mCAAK,IAAL,C;MAAvB,Y;K;IAIyC,sC;8BAAwD,O;;K;;IACjG,yC;MAAA,6D;MAAuB,gCAAK,IAAL,C;MAA vB,Y;K;IAGkD,sD;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IAC5G,kD;MAAA,sE;MAAuB,yCAAK,IA AL,EA AW,IAAX,C;MAAvB,Y;K;IACA,6D;MAAA,sE;MAAuC,yCAAK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IA CA,2D;MAAA,sE;MAAwC,yCAAK,SAAL,EAAGB,KAAhB,C;MAAxC,Y;K;IAG0D,8D;MAA0D,4BAAiB,OAAj B,EAA0B,KAA1B,C;;K;;IACpH,0D;MAAA,8E;MAAuB,iDAAK,IAAL,EA AW,IAAX,C;MAAvB,Y;K;IACA,qE; MAAA,8E;MAAuC,iDAAK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IACA,mE;MAAA,8E;MAAwC,iDAAK,SAAL,E AAGB,KAAhB,C;MAAxC,Y;K;6FCIGJ,yB;MAEI,OAAG,GAAG,CAAC,QAAD,C;K;mFAGV,oB;MAEI,OAAJ,G AAI,GAAE,G;K;6ETVN,a;MAK8C,cAAvC,C;K;6EChP,Y;MAG+C,S;K;IA6B/C,2B;MAG4D,0BA Ae,WAAf,C;K ;IAE5D,mC;MAIwF,0BA Ae,WAAf,C;K;IAExF,mC;MAKwE,0BA Ae,WAAf,C;K;IAGxE,4B;MAI8D,Q;MAH1D, aAAkB,GAAL,O;MACtB,aAAkB,GAAL,O;MACtB,YAAiB,C;MACjB,OAAO,QAAQ,MAAR,IAAkB,QAAQ,MA AjC,C;QAAyC,IAAI,KAAJ,IAAa,IAAI,YAAJ,EAAL,oBAAJ,O;;MACtD,OAAO,G;K;IAIX,wD;MAMuC,Q;MALn C,aAAa,MAAO,OAAM,CAAN,EAAS,OAAT,C;MA0BpB,IAZBc,MAyBL,OAAL,KAAkB,SAAtB,C;QAZBsB,M A0BIB,UA1BU,MA0BS,O;;MAZbVb,YAAiB,MAAO,O;MACxB,IAAI,UAAU,KAAAd,C;QACl,gBAAgB,O;QACH B,OAAO,QAAQ,OAAf,C;UAAwB,OAAO,YAAP,EA AO,oBAAP,UAAkB,Y;;;MAE9C,OAAO,M;K;IAGX,gD;M AKoB,UAAmB,M;MAJnC,aAAa,KAAM,Q;MACnB,MAAO,OAAP,IAAiB,UAAW,K;MAc5B,IAbc,KAA L,OAAL ,KAAkB,SAAtB,C;QAbqB,MACjB,UAdU,KAcS,O;;MabvB,YAAiB,KAAM,O;MACP,4B;MAAhB,OAAGB,cAA hB,C;QAAGB,yB;QAAY,OAAO,cAAP,EA AO,sBAAP,YAAkB,O;;MAC9C,OAAO,M;K;IAGX,yD;MAEoB,UAA gB,M;MADhC,YAA Y,U;MACI,4B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAY,IAAI,cAAJ,EAAL,sBAAJ,YA Ae,O;;MAC3C,OAAO,G;K;oFAGX,oB;MACI,IAAI,IAAK,OAAL,KAAkB,SAAtB,C;QACI,YAAc,IAAK,O;;K;0 EAI3B,wB;MAA+D,OAAA,MAAa,QAAO,GAAP,EAAY,OAAZ,C;K;IS/F5E,mC;MAOI,kBAAkB,MAAa,eAAc, SAAd,C;MAC/B,iBAAiB,MAAa,eAAc,IAAd,C;MAC9B,OAAW,gBAAe,UAA nB,GAA+B,SAA/B,GAAYC,CAA C,S;K;0ECURd,2B;MAKyE,OAAA,MAAa,gBAAe,IAAf,C;K;4EAYbTF,2B;MAKsE,OAAA,MAAa,eAAc,IAAd,C ;K;kEAGnF,qB;MACgD,OAAA,MAAa,KAAK,UAA S,GAAT,EAAC,IAAd,C;K;wEACHc,qB;MAAQ,OAAK,SA AY,a;K;0EACxB,qB;MAAQ,OAAK,SAAY,c;K;IC3D5D,0D;MAGI,OAAO,I;K;ICHX,sC;MAMsD,OAAA,SAAY, UAAS,WAAW,KAA X,CAAT,C;K;IhDKIE,uC;MhBynBW,Q;MAAA,IgBnnBgB,KhBmnBZ,IAAS,CAAT,IgBnnB Y,KhBmnBE,IAAS,wBAA3B,C;QAAA,OAA sC,UgBnnBtB,KhBmnBsB,C;;QgBnnBb,MAAM,8BAA0B,iCAAuB, gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;MhB0nBW,Q;MAAA,IgBpnBgB,KhBonBZ,IAAS,CAAT,IgBpnBY, KhBonBE,IAAS,0BAA3B,C;QAAA,OAA sC,UgBpnBtB,KhBonBsB,C;;QgBpnBb,MAAM,8BAA0B,iCAAuB,gB AAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;MhB2nBW,Q;MAAA,IgBrnBgB,KhBqnBZ,IAAS,CAAT,IgBrnBY,Kh



AoG,UAApG,EAAgH,QAaHh,C;QACA,OAAO,W;O;KArBX,C;wFAwBA,yB;MAAA,8C;MAAA,kF;QAmBsE,i  
C;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QACnI,UAAU,SAAV,EAA2C  
,WAA3C,EAAmF,iBAAnF,EAAgG,UAAiG,EAAkH,QAaIH,C;QACA,OAAO,W;O;KArBX,C;uFAwBA,yB;MA  
AA,8C;MAAA,kF;QAmBwE,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;  
QACrI,UAAU,SAAV,EAA4C,WAA5C,EAAqF,iBAArF,EAAwG,UAAxG,EAAoH,QAAPh,C;QACA,OAAO,W;  
O;KArBX,C;yFAwBA,yB;MAAA,8C;MAAA,kF;QAmBoE,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAAkB,C;Q  
AAG,wB;UAAA,WAAgB,gB;QACvI,UAAU,SAAV,EAA6C,WAA7C,EAAuF,iBAAvF,EAA0G,UAA1G,EAAhH,  
QAAtH,C;QACA,OAAO,W;O;KArBX,C;yFAwBA,yB;MAAA,8C;MAAA,kF;QAmBoE,iC;UAAA,oBAAYB,C;Q  
AAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QACjI,UAAU,SAAV,EAA0C,WAA1C,EAAiF,iBAAj  
F,EAAoG,UAApG,EAAgH,QAaHh,C;QACA,OAAO,W;O;KArBX,C;oFAwBA,qB;MAOI,OAAY,SAAY,Q;K;oF  
AG5B,qB;MAOI,OAAY,SAAY,Q;K;oFAG5B,qB;MAOI,OAAY,SAAY,Q;K;qFAG5B,qB;MAOI,OAAY,SAAY,  
Q;K;IAG5B,8B;MAMW,WAAS,W;MAAT,YAA2B,SAAY,Q;MkCl7B9C,eAAiB,I;MICK7BjB,OkCj7BO,K;K;qFl  
Co7BX,qB;MAOI,OAAY,SAAY,Q;K;qFAG5B,qB;MAOI,OAAY,SAAY,Q;K;IAG5B,8B;MAMW,WAAS,c;MAA  
T,YAA8B,SAAY,Q;MkC/8BjD,eAAiB,I;MIC+8BjB,OkC98BO,K;K;IICi9BX,8B;MAMW,WAAS,W;MAAT,YAA  
2B,SAAY,Q;MkCx9B9C,eAAiB,I;MICw9BjB,OkCv9BO,K;K;IIC09BX,uC;MD5oCI,IAAI,ECspCI,WAAW,CDtp  
Cf,CAAJ,C;QACI,cCqpCoB,0C;QDppCpB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MCqpCV,OAAO,SAAS,SAAT,  
EAAe,cAAU,OAAV,CAAF,C;K;IAGX,uC;MD1pCI,IAAI,ECqCI,WAAW,CDpqCf,CAAJ,C;QACI,cCmqCoB,0C  
;QDlqCpB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MCmqCV,OAAO,SAAS,SAAT,EAAe,eAAW,OAAZ,CAAF,C;K  
;IAGX,uC;MDxqCI,IAAI,ECkrCI,WAAW,CDlrCf,CAAJ,C;QACI,cCirCoB,0C;QDhrCpB,MAAM,gCAAYB,OAA  
Q,WAAjC,C;;MCirCV,OAAO,SAAS,SAAT,EAAe,eAAS,OAAT,CAAF,C;K;IAGX,uC;MDtrCI,IAAI,ECgsCI,WA  
AW,CDhsCf,CAAJ,C;QACI,cC+rCoB,0C;QD9rCpB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MC+rCH,WAAS,W;M  
AAT,YAAsB,gBAAGB,SAAhB,EAAhB,OAAtB,K;MkChhC7B,eAAiB,I;MICghCjB,OkC/gCO,K;K;IICkhCX,uC;  
MDpsCI,IAAI,EC8sCI,WAAW,CD9sCf,CAAJ,C;QACI,cC6sCoB,0C;QD5sCpB,MAAM,gCAAYB,OAAQ,WAAj  
C,C;;MC6sCV,OAAO,SAAS,SAAT,EAAe,iBAAW,OAAZ,CAAF,C;K;IAGX,uC;MDltCI,IAAI,EC4tCI,WAAW,C  
D5tCf,CAAJ,C;QACI,cC2tCoB,0C;QD1tCpB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MC2tCV,OAAO,SAAS,SA  
T,EAAe,iBAAY,OAAs,CAAF,C;K;IAGX,uC;MDhuCI,IAAI,EC0uCI,WAAW,CD1uCf,CAAJ,C;QACI,cCyuCoB,  
0C;QDxuCpB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MCyuCH,WAAS,c;MAAT,YAAyB,gBAAGB,SAAhB,EAAh  
B,OAAtB,EAA+B,KAA/B,C;MkC1jChC,eAAiB,I;MIC0jCjB,OkCzjCO,K;K;IIC4jCX,uC;MD9uCI,IAAI,ECwvCI,  
WAAW,CDxvCf,CAAJ,C;QACI,cCuvCoB,0C;QDtvCpB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MCuvCH,WAAS,  
W;MAAT,YAAsB,SAAS,SAAT,EAAe,iBAAU,OAAV,CAAF,C;MkCxc7B,eAAiB,I;MICwkCjB,OkCvkCO,K;K;  
IIC0kCX,uC;MD5vCI,IAAI,ECuwCI,WAAW,CDvwCf,CAAJ,C;QACI,cCswCoB,0C;QDrwCpB,MAAM,gCAAYB  
,OAAQ,WAAjC,C;;MCswCV,OAAO,gBAAGB,SAAhB,EAAhB,OAAtB,EAA+B,IAA/B,C;K;IAGX,sD;MAWI,oC  
AAa,2BAAkB,SAaIB,EAA6B,OAA7B,EAAc,gBAAtC,C;MACb,OAAY,SAAY,OAAM,SAAN,EAAiB,OAAjB,  
C;K;IAG5B,sD;MAUI,oCAAA,2BAAkB,SAaIB,EAA6B,OAA7B,EAAc,gBAAtC,C;MACb,OAAY,SAAY,OA  
M,SAAN,EAAiB,OAAjB,C;K;IAG5B,sD;MAUI,oCAAA,2BAAkB,SAaIB,EAA6B,OAA7B,EAAc,gBAAtC,C;M  
ACb,OAAY,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;K;IAG5B,sD;MAUI,oCAAA,2BAAkB,SAaIB,EAA6B,OA  
A7B,EAAc,gBAAtC,C;MACb,OAAY,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;K;IAG5B,sD;MAUI,oCAAA,2BA  
AkB,SAaIB,EAA6B,OAA7B,EAAc,gBAAtC,C;MACN,WAAS,W;MAAT,YAA2B,SAAY,OAAM,SAAN,EAAi  
B,OAAjB,C;MkC9pC9C,eAAiB,I;MIC8pCjB,OkC7pCO,K;K;IICgqCX,sD;MAUI,oCAAA,2BAAkB,SAaIB,EAA6  
B,OAA7B,EAAc,gBAAtC,C;MACb,OAAY,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;K;IAG5B,sD;MAUI,oCAA  
a,2BAAkB,SAaIB,EAA6B,OAA7B,EAAc,gBAAtC,C;MACb,OAAY,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;K  
;IAG5B,uD;MAUI,oCAAA,2BAAkB,SAaIB,EAA6B,OAA7B,EAAc,gBAAtC,C;MACN,WAAS,c;MAAT,YAA8  
B,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;MkCxsCjD,eAAiB,I;MICwsCjB,OkCvsCO,K;K;IIC0sCX,uD;MAUI,o  
CAAA,2BAAkB,SAaIB,EAA6B,OAA7B,EAAc,gBAAtC,C;MACN,WAAS,W;MAAT,YAA2B,SAAY,OAAM,S  
AAN,EAAiB,OAAjB,C;MkCttC9C,eAAiB,I;MICstCjB,OkCrtCO,K;K;IICwtCX,wD;MAWgD,yB;QAAA,YAAiB,  
C;MAAG,uB;QAAA,UAAe,gB;MAC/E,oCAAA,2BAAkB,SAaIB,EAA6B,OAA7B,EAAc,gBAAtC,C;MiDz3CD,  
ejD03CD,OiD13CC,EjD03CQ,SiD13CR,EjD03CmB,OiD13CnB,C;K;IjD63ChB,wD;MAWgD,yB;QAAA,YAAiB,  
C;MAAG,uB;QAAA,UAAe,gB;MAC/E,oCAAA,2BAAkB,SAaIB,EAA6B,OAA7B,EAAc,gBAAtC,C;MiDz4CD,





gD;MAagC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MAC/D,oCAAA,2BAakB,SAAI B,EAA6B,OAA7 B,EAAsC,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAApB,C;MACvB,KAAT,QAAS,C;K;IAGb, gD;MAaiC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACHe,oCAAA,2BAakB,SAAI B,EAA6B,OAA7 B,EAAsC,gBAAtC,C;MACb,gBAAc,SAAd,EAA8C,SAAY,UAAS,SAAT,EAAoB,OAApB,C;MACvB,KAAT,QAAS,C;K;IAGJ,gD;MAa kC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACjE,oCAAA,2BAakB,SAAI B,EAA6B,OAA7B,EAAsC ,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAApB,C;MACvB,KAAT,QAAS,C;K;IAGb,gD;MAam C,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACIE,oCAAA,2BAakB,SAAI B,EAA6B,OAA7B,EAAsC, gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAApB,C;MACvB,KAAT,QAAS,C;K;IAGb,gD;MAaiC ,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACHe,oCAAA,2BAakB,SAAI B,EAA6B,OAA7B,EAAsC,g BAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAApB,C;MACvB,KAAT,QAAS,C;K;IFAGb,iC;MkD1+ DgB,elDi/DD,UkDj/DC,C;K;iFIDo/DhB,iC;MkDp/DgB,elD2/DD,UkD3/DC,C;K;iFID8/DhB,iC;MkD9/DgB,elDqg ED,UkDrgEC,C;K;iFIDwgEhB,iC;MkDxgEgB,elD+gED,UkD/gEC,C;K;iFIDkhEhB,iC;MkDlhEgB,elDyhED,UkD zhEC,C;K;iFID4hEhB,iC;MkD5hEgB,elDmiED,UkDniEC,C;K;iFIDsiEhB,iC;MkDtiEgB,elD6iED,UkD7iEC,C;K;I lDgiEhB,yC;MAMI,IAAI,mBAAO,CAAX,C;QAAC,gBAAc,SAAd,EAAoB,UAApB,C;K;IAGIB,+D;MAa0E,yB;Q AAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACzG,oCAAA,2BAakB,SAAI B,EAA6B,OAA7B,EAAsC,gBAAt C,C;MACb,gBAAc,SAAd,EAAoB,SAApB,EAA+B,OAA/B,EAAwC,UAAxC,C;K;IAGJ,mC;MAII,OAAO,EAAS, MAAM,MAAK,SAAL,C;K;IAG1B,mC;MAII,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAG1B,mC;MAII,OAA O,EAAS,MAAM,MAAK,SAAL,C;K;IAG1B,mC;MAII,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAG1B,mC;M AII,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAG1B,mC;MAII,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAG1 B,mC;MAII,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAOH,kD;MAAA,wB;QAAW,qCAAK,KAAL,E;O;K;IAJI C,oC;MAII,OAAO,iBAAM,gBAAN,EAAY,gCAAZ,C;K;ImDnpEX,oB;MAAA,wB;MAEI,6B;MACA,gC;MAKu B,UAAT,MAAS,EAAT,MAAS,EAAT,M;MAFV,eAAe,kE;MACf,iBAAiB,eAAS,GAAT,C;MACE,sBAAT,QAAS ,C;MAAT,mB;MAAA,kB;MAAA,kB;MAAV,8C;QACI,WAAW,oBAAS,CAAT,CrC4BuB,IqC5BIC,IAA+B,C;M AInC,qBAaQb,48C;MACrB,WAAW,mBAAmB,cAAnB,EAAmC,UAAnC,EAA+C,IAA/C,C;MACX,YAAY,eAA S,IAAK,OAAL,GAAY,CAAZ,IAAT,C;MACZ,0BAAU,IAAV,e;QACI,MAAM,MAAI,CAAJ,IAAN,IAAe,MAAM ,GAAN,IAAW,KAAL,GAAL,CAAX,I;MAEnB,yBAAoB,K;MAGpB,oBAAoB,m/D;MACpB,4BAAuB,mBAAm B,aAAnB,EAakC,UAAIC,EAA8C,IAA9C,C;K;IAvB/B,gC;MAAA,+B;QAAA,c;MAAA,wB;K;IA2BA,qC;MA KkB,IAJP,I;MACH,WAAO,EAAP,C;QA Ae,W;WAcF,WAAO,IAAP,C;QAAGB,OAAL,CAAC,KAAO,CAAR,MA Ac,CAAIB,GAAqB,QAAS,CAA9B,GAAqC,OAAS,E;QAE1D,QAAM,KAAL,CAAL,IAAN,C;eACI,C;YAAK,e AAS,E;YAAAd,K;eACA,C;YAAK,OAAC,QAAS,CAAV,GAAiB,E;YAAAtB,K;YACQ,cAAS,E;YAHrB,K;MAJR, W;K;IAYJ,qC;MAII,SAAS,SrCPiC,I;MqCS1C,YAAY,kBAakB,sBAAS,kBAA3B,EAA8C,EAA9C,C;MACZ,YA AY,sBAAS,kBAAT,CAA2B,KAA3B,C;MACZ,WAAW,sBAAS,qBAAT,CAA8B,KAA9B,C;MACX,YAAY,kBA AkB,IAAIB,EAAwB,KAAL,IAAxB,C;MAEZ,OAAW,UAAS,EAAb,GAAyC,mDAAzC,GAAoD,K;K;IAG /D,8D;MAKiB,UAIE,M;MARf,aAAa,eAAS,YAAT,C;MACb,YAAY,C;MACZ,UAAU,C;MACV,YAAY,C;MAC C,yB;MAAb,OAAa,cAAb,C;QAAa,iC;QACT,aAAa,WAAW,IrCvBc,IqCuBzB,C;QACb,MAAM,MAAQ,CAAC,S AAW,EAAZ,KAAAsB,K;QACpC,IAAI,SAAS,EAAb,C;UACI,OAAO,cAAP,EAAO,sBAAP,YAAkB,G;UACIB,M AAM,C;UACN,QAAQ,C;UAER,gBAAS,CAAT,I;MAGR,OAAO,M;K;ICIEX,+B;MAII,eAAe,CAAC,iBAAO,C AAP,IAAD,IAAa,CAAb,I;MACf,IAAI,WAAW,CAAF,C;QAAkB,M;MACIB,mBAAmB,2B;MACnB,iBAAc,CAA d,WAAiB,QAAjB,U;QACI,UAAU,sBAAK,KAAL,C;QACV,sBAAK,KAAL,EAAc,sBAAK,YAAL,CAAd,C;QAC A,sBAAK,YAAL,EAAqB,GAAR,C;QACA,mC;K;IjDbR,wB;MAOI,OAAW,oBAAK,CAAL,MAAJ,GAAY,CA AZ,GAAM,B,C;K;mFAG9B,yB;MAkBA,iB;MAIBA,uB;QAMI,OakBO,MAAO,KAI BC,CAkBD,EAIBY,CAkBZ, C;O;KAXBIB,C;mFASA,yB;MASA,iB;MATA,uB;QAMI,OASO,MAAO,KATC,CASD,EATY,CASZ,C;O;KAFiB, C;mFASA,yB;MAAA,iB;MAAA,uB;QAMI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KANIB,C;mFASA,g B;MAMI,OAAW,kBAAK,CAAL,MAAJ,GAAY,CAAZ,GAAM,B,C;K;mFAG9B,yB;MAAA,iB;MAAA,uB;QAQI, OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KARIB,C;mFAWA,yB;MAAA,iB;MAAA,uB;QAQI,OAAO,MA AO,KAAI,CAAJ,EAAO,CAAP,C;O;KARIB,C;IAWA,2B;MAOI,OAAO,SAAM,CAAN,EAAS,SAAM,CAAN,EA AS,CAAT,CAAT,C;K;mFAGX,yB;MAAA,iB;MAAA,0B;QAMI,OAAO,MAAO,KAAM,CAAN,EAAiB,CAAjB, EAA4B,CAA5B,C;O;KANIB,C;mFASA,yB;MAAA,iB;MAAA,0B;QAMI,OAAO,MAAO,KAAM,CAAN,EAAiB,

CAAjB,EAA4B,CAA5B,C;O;KANIB,C;mFASA,yB;MAAA,iB;MAAA,0B;QAMI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,EAAU,CAAV,C;O;KANIB,C;mFASA,mB;MAMW,UAAe,CAPeX,iBAoEc,CAPeD,MAAJ,GAoEe,CAPeF,GAoEkB,C;MAAzB,OAAa,CAPeF,iBAAK,GAAL,MAAJ,GAoEM,CAPeN,GAAMb,G;K;mFAuE9B,yB;MAAA,iB;MAAA,0B;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,EAAU,CAAV,C;O;KARIB,C;mFAWA,yB;MAAA,iB;MAAA,0B;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,EAAU,CAAV,C;O;KARIB,C;IAWA,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAAM,SAAM,GAAN,EA AW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAxHV,MAAO,KAwHe,GAxHf,EAwHoB,CxHpB,C;;MAyHd,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAIIV,MAAO,KAKIe,GAlIf,EAKIoB,CAlIpB,C;;MAMId,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MA5IV,MAAO,KA4Ie,GA5If,EA4IoB,CA5IpB,C;;MA6Id,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAuB,UAAM,G;QAAZ,MA7IN,oBA6IuB,CA7IvB,MAAJ,GAAY,GA AZ,GA6I2B,C;;MACIC,OAAO,G;K;IAGX,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MA9IV,MAAO,KA8Ie,GA9If,EA8IoB,CA9IpB,C;;MA+Id,OAAO,G;K;IAGX,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MA/IV,MAAO,KA+Ie,GA/If,EA+IoB,CA/IpB,C;;MAGJd,OAAO,G;K;IAGX,wB;MAOI,OAAW,oBAAK,CAAL,MAAJ,GAAY,CAAZ,GAAMb,C;K;mFAG9B,yB;MAkBA,iB;MAiBA,uB;QAMI,OAKBO,MAAO,KAIBC,CAkBD,EAiBY,CAkBZ,C;O;KAXBiB,C;mFASA,yB;MASA,iB;MATA,uB;QAMI,OASO,MAAO,KAATC,CASD,EATY,CASZ,C;O;KAlfB,C;mFASA,yB;MAAA,iB;MAAA,uB;QAMI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KANIB,C;mFASA,gB;MAMI,OAAW,kBAAK,CAAL,MAAJ,GAAY,CAAZ,GAAMb,C;K;mFAG9B,yB;MAAA,iB;MAAA,uB;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KARIB,C;mFAWA,yB;MAAA,iB;MAAA,uB;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KARIB,C;IAWA,2B;MAOI,OAAO,SAA M,CAAN,EAAS,SAAM,CAAN,EAAS,CAAT,CAAT,C;K;mFAGX,yB;MAAA,iB;MAAA,0B;QAMI,OAAO,MAAO,KAAM,CAAN,EA AiB,CAAjB,EAA4B,CAA5B,C;O;KANIB,C;mFASA,yB;MAAA,iB;MAAA,0B;QAMI,OAAO,MAAO,KAAM,CAAN,EA AiB,CAAjB,EAA4B,CAA5B,C;O;KANIB,C;mFASA,yB;MAAA,iB;MAAA,0B;QAMI,OAAO,MAAO,KAAM,CAAN,EA AiB,CAAjB,EAA4B,CAA5B,C;O;KANIB,C;mFASA,mB;MAMW,UAAe,CAPeX,iBAoEc,CAPeD,MAAJ,GAoEe,CAPeF,GAoEkB,C;MAAzB,OAAa,CAPeF,iBAAK,GAAL,MAAJ,GAoEM,CAPeN,GAAMb,G;K;mFAuE9B,yB;MAAA,iB;MAAA,0B;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,EAAU,CAAV,C;O;KARIB,C;mFAWA,yB;MAAA,iB;MAAA,0B;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,EAAU,CAAV,C;O;KARIB,C;IAWA,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAAM,SAAM,GAAN,EA AW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAxHV,MAAO,KAwHe,GAxHf,EAwHoB,CxHpB,C;;MAyHd,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAIIV,MAAO,KAKIe,GAlIf,EAKIoB,CAlIpB,C;;MAMId,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MA5IV,MAAO,KA4Ie,GA5If,EA4IoB,CA5IpB,C;;MA6Id,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAuB,UAAM,G;QAAZ,MA7IN,oBA6IuB,CA7IvB,MAAJ,GAAY,GA AZ,GA6I2B,C;;MACIC,OAAO,G;K;IAGX,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MA9IV,MAAO,KA8Ie,GA9If,EA8IoB,CA9IpB,C;;MA+Id,OAAO,G;K;IAGX,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MA/IV,MAAO,KA+Ie,GA/If,EA+IoB,CA/IpB,C;;MAGJd,OAAO,G;K;IkDvaX,iB;MAAA,qB;MAEI,0BAA0B,gBACtB,EADsB,EACd,IADc,EACN,IADM,EACE,IADF,EACU,IADV,EACKb,IADIB,EAC0B,IAD1B,EACKc,IADIC,EAC0C,IAD1C,EACKd,IADID,EAC0D,IAD1D,EACKe,IADIE,EAC0E,IAD1E,EACKf,IADIF,EAC0F,IAD1F,EACKg,IADIG,EAC0G,IAD1G,EACKh,IADIH,EAC0H,IAD1H,EACKI,IADII,EAETb,IAFsB,EAEd,IAFc,EAEN,IAFM,EAEE,IAFF,EAEU,IAFV,EAEB,IAFIB,EA E0B,IAF1B,EAekC,IAFIC,EAEOC,IAF1C,EAekD,KAFID,EAEOd,KAF1D,EAekE,KAFIE,EAEOE,KAF1E,EAekF,KAFIF,EAEOf,KAF1F,EAekG,KAFIG,EAEOg,KAF1G,E;K;;IAF9B,6B;MAAA,4B;QAAA,W;;MAAA,qB;K;IAQA,0C;MAKI,aAAa,C;MACb,UAAU,KAAM,OAAN,GA Aa,CAAb,I;MACV,aAAa,E;MACb,YAA Y,C;MACZ,OAAO,UAAU,GAAjB,C;QACI,SAAS,CAAC,SAAS,GAAT,IAAD,IAAiB,CAAjB,I;QACT,QAAQ,MAAM,MAA

N,C;QACR,IAAI,SAAS,KAAb,C;UACI,SAAS,SAAS,CAAT,I;aACR,IAAI,WAAU,KAAc,C;UACD,OAAO,M;;U  
AEP,MAAM,SAAS,CAAT,I;;MAEd,OAAO,UAAc,SAAS,KAAb,GAAoB,CAApB,GAA2B,CAArC,K;K;IAGX,m  
C;MAKI,SAAS,SvCEiC,I;MuCD1C,YAAY,kBAakB,mBAAM,mBAAxB,EAAoC,EAApC,C;MACZ,WAAW,KA  
AK,mBAAM,mBAAN,CAAiB,KAAjB,CAAL,I;MACX,OAAW,OAAO,EAAx,GAAe,IAAf,GAAyB,E;K;IAGpC,  
gC;MAII,OAAO,6BAAoB,C;K;IC7C/B,kB;MAAA,sB;MAEI,6B;MACA,8B;MACA,gC;MAKuB,UAAT,MAAS,E  
AAT,MAAS,EAAT,M;MAFV,eAAe,kE;MACf,iBAaiB,eAAS,GAAT,C;MACE,sBAAT,QAAS,C;MAAT,mB;MA  
AA,kB;MAAA,kB;MAAV,8C;QACI,WAAW,oBAAS,CAAT,CxC2BuB,IwC3BIC,IAA+B,C;;MAInC,qBAAqB,s  
W;MACrB,WAAW,mBAAmB,cAAnB,EAAmC,UAAnc,EAA+C,GAA/C,C;MACX,YAAY,eAAS,IAAK,OAAc,  
C;MACZ,0BAAU,IAAV,e;QACI,IAAI,QAAC,CAAT,C;UAAy,MAAM,GAAN,IAAW,KAAK,GAAL,C;;UACIB,  
MAAM,GAAN,IAAW,MAAM,MAAI,CAAJ,IAAN,IAAe,KAAK,GAAL,CAAf,I;;MAEpB,yBAAoB,K;MAGpB,k  
BAakB,0U;MACIB,0BAAqB,mBAAmB,WAAnc,EAAgC,UAAhC,EAA4C,GAA5C,C;MAGrB,oBAAoB,i8B;M  
ACpB,4BAAuB,mBAAmB,aAAnB,EAAkC,UAAIC,EAA8C,GAA9C,C;K;;;IA7B/B,8B;MAAA,6B;QAAA,Y;;MA  
AA,sB;K;IAiCA,iC;MAII,OAAO,6BAAmB,C;K;IAG9B,oC;MAIW,wCAAmB,C;MAAnB,U;QAA6B,wBxCpM,a  
wCON,C;;MAApC,W;K;IAGJ,oC;MAIW,wCAAmB,C;MAAnB,U;QAA6B,wBxCdM,awCeN,C;;MAApC,W;K;IA  
GJ,kC;MAQI,SAAS,SxCzBiC,I;MwC0B1C,YAAY,kBAakB,oBAAO,kBAAzB,EAA4C,EAA5C,C;MAEZ,iBAai  
B,oBAAO,kBAAP,CAAYB,KAAzB,C;MACjB,eAAe,aAAa,oBAAO,mBAAP,CAA0B,KAA1B,CAAb,GAAgD,C  
AAhD,I;MACf,WAAW,oBAAO,qBAAP,CAA4B,KAA5B,C;MAEX,IAAI,KAAK,QAAT,C;QACI,OAAO,C;;MA  
GX,kBAakB,OAAS,C;MAE3B,IAAI,gBAae,CAAnB,C;QACI,YAAY,C;QACZ,gBAAgB,U;QACHB,aAAU,CAA  
V,OAAa,CAAb,M;UACI,yBAAc,QAAS,KAAV,GAAqB,GAAIC,K;UACA,IAAI,YAAY,EAAhB,C;YACI,OAAO,  
C;;UAEX,gBAAS,CAAT,I;UACA,yBAAc,QAAS,KAAV,GAAqB,GAAIC,K;UACA,IAAI,YAAY,EAAhB,C;YA  
CI,OAAO,C;;UAEX,gBAAS,CAAT,I;;QAEJ,OAAO,C;;MAGX,IAAI,QAAQ,CAAZ,C;QACI,OAAO,W;;MAGX,e  
AAgB,KAAK,UAAL,I;MACHB,cAAgB,QAAQ,EAZ,GAAkB,WAAW,CAAX,IAAIB,GAAoC,Q;MACHD,OAA  
Q,SAAU,IAAI,OAAJ,IAAV,CAAD,GAA2B,C;K;ICnGtC,0B;MAAA,8B;MACI,+BAA+B,gBAC3B,GAD2B,EAC  
nB,GADmB,EACX,GADW,EACH,GADG,EACK,GADL,EACa,GADb,EACqB,GADrB,EAC6B,IAD7B,EACqC,I  
ADrC,EAC6C,IAD7C,EACqD,IADrD,EAC6D,IAD7D,EACqE,IADrE,EAC6E,IAD7E,EACqF,IADrF,EAC6F,KA  
D7F,EACqG,KADrG,EAC6G,KAD7G,EACqH,KADrH,EAC6H,KAD7H,E;MAG/B,gCAAgC,gBAC5B,CAD4B,E  
ACzB,CADyB,EACtB,CADsB,EACnB,CADmB,EACHB,CADgB,EACb,CADa,EACV,CADU,EACP,EADO,EAC  
H,CADG,EACA,EADA,EACI,CADJ,EACO,CADP,EACU,EADV,EACc,EADd,EACkB,EADIB,EACsB,CADtB,E  
ACyB,CADzB,EAC4B,CAD5B,EAC+B,CAD/B,EACkC,CADIC,E;K;;;IAJpC,sC;MAAA,qC;QAAA,oB;;MAAA,8  
B;K;IASA,qC;MACI,YAAY,kBAakB,4BAAe,wBAAjC,EAakD,SAaID,C;MACZ,OAAO,SAAS,CAAT,IAAc,aA  
AO,4BAAe,wBAAf,CAA+B,KAA/B,IAAwC,4BAAe,yBAaf,CAAgC,KAAhC,CAAxC,IAAP,C;K;ICXzB,qC;MA  
CI,OAAe,IAAR,8BAAgB,IAAhB,KACY,IAAR,8BAAgB,IADpB,C;K;ICCX,wC;MxCiBW,Q;MAAA,IwCXgB,K  
xCWZ,IAAS,CAAT,IwCXY,KxCWE,IAAS,2BAA3B,C;QAAA,OAAsC,qBwCXtB,KxCWsB,C;;QwCXb,MAAM,  
8BAA0B,mCAAYB,gBAAzB,MAA1B,C;;MAAtC,W;K;ICRJ,sC;MAEI,WAAW,S5CmC+B,I;M4CjC1C,IAAY,G  
AAR,oBAAgB,GAAhB,KAAkC,GAAR,oBAAgB,GAA1C,CAAJ,C;QACI,OAA8B,OAAtB,KAAK,CAAC,OAAO  
,CAAP,IAAD,IAAa,CAAb,IAAL,KAAsB,C;;MAGIC,IAAY,IAAR,oBAAgB,IAAhB,KAAkC,IAAR,oBAAgB,IA  
A1C,CAAJ,C;QACI,OAAO,S;;MAEX,OAAO,wB;K;ICPX,wC;MpCqTe,WoC7SY,KpC6SZ,IAAS,C;MAAT,S;QA  
Ac,OoC7SF,KpC6SE,IAyHT,gBAAR,iBAAQ,C;;MAZgHT,U;MAAA,S;QAAA,SAAsC,sBoC7StB,KpC6SsB,C;;  
QoC7Sb,MAAM,8BAA0B,iCAAuB,cAAvB,MAA1B,C;;MAAtC,a;K;IAGJ,wC;MpCsTe,WoC9SY,KpC8SZ,IAAS  
,C;MAAT,S;QAAC,OoC9SF,KpC8SE,IAqgHT,gBAAR,iBAAQ,C;;MArgHT,U;MAAA,S;QAAA,SAAsC,sBoC9St  
B,KpC8SsB,C;;QoC9Sb,MAAM,8BAA0B,iCAAuB,cAAvB,MAA1B,C;;MAAtC,a;K;IAGJ,wC;MpCuTe,WoC/SY,  
KpC+SZ,IAAS,C;MAAT,S;QAAC,OoC/SF,KpC+SE,IAigHT,gBAAR,iBAAQ,C;;MAjgHT,U;MAAA,S;QAAA,SA  
AsC,sBoC/StB,KpC+SsB,C;;QoC/Sb,MAAM,8BAA0B,iCAAuB,cAAvB,MAA1B,C;;MAAtC,a;K;IAGJ,wC;MpC  
wTe,WoChTY,KpCgTZ,IAAS,C;MAAT,S;QAAC,OoChTF,KpCgTE,IA6/GT,gBAAR,iBAAQ,C;;MA7/GT,U;MA  
AA,S;QAAA,SAAsC,sBoChTtB,KpCgTsB,C;;QoChTb,MAAM,8BAA0B,iCAAuB,cAAvB,MAA1B,C;;MAAtC,a;  
K;IASO,6C;MAAA,8B;MAAS,uB;K;8FACW,Y;MAAQ,OAAA,gBAAY,K;K;+CAC3C,Y;MAAkC,OAAA,gBAA  
Y,U;K;sDAC9C,mB;MAAgD,OAAA,gBAAY,gBAAS,OAAT,C;K;mDAC5D,iB;MACI,oCAAa,2BAakB,KAAIB,  
EAAyB,SAAzB,C;MACb,OAAO,6BAAY,KAAZ,C;K;qDAEX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,OAAJ,

C;QAAgC,OAAO,E;MACvC,OpC0rBO,UoC1rBA,gBpC0rBR,QAAQ,EoC1rBoB,OxEgOF,KoC0dlB,C;K;yDoCxr BX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,OAAJ,C;QAAgC,OAAO,E;MACvC,OpC66BO,coC76BA,gBpC66 BR,QAAQ,EoC76BwB,OxE2NN,KoCktBIB,C;K;;IoCn8BnB,6B;MAMI,4C;K;IA2BO,6C;MAAA,8B;MAAS,uB; K;8FACW,Y;MAAQ,OAAA,gBAAY,K;K;+CAC3C,Y;MAAkC,OAAA,gBAAY,U;K;sDAC9C,mB;MAAiD,OAA A,gBAAY,gBAAS,OAAT,C;K;mDAC7D,iB;MACI,oCAAA,2BAAkB,KAAIB,EAAyB,SAAZB,C;MACb,OAAO,6 BAAY,KAAZ,C;K;qDAEX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,QAAJ,C;QAAiC,OAAO,E;MACxC,OpCy qBO,UoCzqBA,gBpCyqBR,QAAQ,EoCzqBoB,OvDgNA,KmBydpB,C;K;yDoCvqBX,mB;MAES,Q;MAAL,IAAI, eAAC,0EAAD,QAAJ,C;QAAiC,OAAO,E;MACxC,OpC45BO,coC55BA,gBpC45BR,QAAQ,EoC55BwB,OvD2M J,KmBitBpB,C;K;;IoC17BnB,6B;MAMI,4C;K;IA2BO,6C;MAAA,8B;MAAS,uB;K;8FACW,Y;MAAQ,OAAA,gB AAY,K;K;+CAC3C,Y;MAAkC,OAAA,gBAAY,U;K;sDAC9C,mB;MAAiD,OAAA,gBAAY,gBAAS,OAAT,C;K; mDAC7D,iB;MACI,oCAAA,2BAAkB,KAAIB,EAAyB,SAAZB,C;MACb,OAAO,6BAAY,KAAZ,C;K;qDAEX,mB ;MAES,Q;MAAL,IAAI,eAAC,0EAAD,QAAJ,C;QAAiC,OAAO,E;MACxC,OpCwpBO,UoCxpBA,gBpCwpBR,Q AAQ,EoCxpBoB,OzEkIA,KqCshBpB,C;K;yDoCtpBX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,QAAJ,C;QAAi C,OAAO,E;MACxC,OpC24BO,coC34BA,gBpC24BR,QAAQ,EoC34BwB,OzE6HJ,KqC8wBpB,C;K;;IoCj6BnB,8 B;MAMI,4C;K;IA2BO,6C;MAAA,8B;MAAS,uB;K;8FACW,Y;MAAQ,OAAA,gBAAY,K;K;+CAC3C,Y;MAAkC ,OAAA,gBAAY,U;K;sDAC9C,mB;MAAkD,OAAA,gBAAY,gBAAS,OAAT,C;K;mDAC9D,iB;MACI,oCAAA,2B AAkB,KAAIB,EAAyB,SAAZB,C;MACb,OAAO,6BAAY,KAAZ,C;K;qDAEX,mB;MAES,Q;MAAL,IAAI,eAAC, 0EAAD,SAAJ,C;QAAkC,OAAO,E;MACzC,OpCuoBO,UoCvoBA,gBpCuoBR,QAAQ,EoCvoBoB,OvEkHE,KmC qhBtB,C;K;yDoCroBX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,SAAJ,C;QAAkC,OAAO,E;MACzC,OpC03BO, coC13BA,gBpC03BR,QAAQ,EoC13BwB,OvE6GF,KmC6wBtB,C;K;;IoCh5BnB,8B;MAMI,4C;K;ICtIJ,qC;MAII, SAAS,S9CgCiC,I;M8C/B1C,OAAa,CAAN,gBAAc,EAAd,KACU,EAAN,gBAAc,EADIB,KAEL,OAAM,GAFV,K AGI,KAAG,IAAL,KACC,OAAM,IAAN,KACS,IAAN,gBAAc,IADjB,KAEG,OAAM,IAFT,IAGG,OAAM,IAHT,I AIG,OAAM,IAJT,IAKG,OAAM,IALT,IAMG,OAAM,KAPV,CAHJ,C;K;;;mCCTP,gB;;K;;ICAJ,wB;K;;IAIA,wB ;K;;IAIA,wB;K;;IAKiC,uB;MAAC,oB;QAAA,OAA0B,E;MAA1B,gB;K;;IAEIC,kB;K;;IAqCqC,sB;MAAC,gB;K;; IAqCN,4B;MAAC,sB;K;;IAEjC,uB;K;;IA8DmC,4B;MAAC,kB;K;;IAEpC,oB;K;;IAmCA,+B;K;;ICvLA,oB;K;;IAI A,wB;K;;oFzDJA,qB;MAKqE,uC8BjtB,E;K;ig9BM/C,yB;MAAA,kD;MAAA,4B;QAQsE,mBAAY,SAAZ,C;O;K ARtE,C;IAUA,iC;MAGI,OAAsB,UAY,QAAvB,KAAmC,SAAY9C,GACe,UAY,UAD3B,GAGI,gBAAGB,UAA hB,C;K;IAGR,qC;MAEI,Y8B3B2C,E;M9B4B3C,eAAe,UAAW,W;MAC1B,OAAO,QAAS,UAAhB,C;QACU,KA AY,MAAK,QAAS,OAAd,C;MACtB,OAAO,K;K;IAGX,8C;MAQc,Q;MANV,IAAI,KAAM,OAAN,GAAa,UAAW ,KAA5B,C;QACI,OAAO,gBAAGB,UAAhB,C;;MAEX,eAAe,UAAW,W;MAC1B,YAAY,C;MACZ,OAAO,QAAS ,UAAhB,C;QACI,MAAM,YAAN,EAAM,oBAAN,UAAiB,QAAS,O;;MAE9B,IAAI,QAAQ,KAAM,OAAB,C;QA CI,MAAM,KAAN,IAAe,I;;MAEnB,OAAO,K;K;IAIX,yB;MAG6C,sBAAY,OAAZ,E;K;wGAE7C,yB;MAAA,+D; MAAA,gC;QAI0B,gBAAf,gB;QAAqB,aJU5B,W;QIVA,OJWO,SIXoC,Q;O;KAJ/C,C;yGAOA,yB;MAAA,4E;MA AA,gE;MAAA,0C;QAI1,qBAaqB,QAArB,C;QAC8B,gBAAvB,eAAa,QAAb,C;QAA6B,aJEpC,W;QIFA,OJGO,SI H4C,Q;O;KALvD,C;IASA,wB;MAG2C,oBAAU,OAAY,E;K;sGAE3C,yB;MAAA,uE;MAAA,gC;QAI8B,gBAAn B,oB;QAAyB,aJXhC,W;QIWA,OJVO,SIUwC,Q;O;KAJnD,C;wGAOA,yB;MAAA,wE;MAAA,0C;QAI5C,gBAA3 B,mBAAiB,QAAjB,C;QAAiC,aJIBxC,W;QIkBA,OJjBO,SIbGd,Q;O;KAJ3D,C;IAQA,qB;MAIuD,oBAAU,IAAV ,E;K;sGAEvD,yB;MAAA,wE;MAAA,gC;QAIiC,gBAAtB,oB;QAA4B,aJhCnC,W;QIGCA,OJ/BO,SI+B2C,Q;O;K AJtD,C;uGAOA,yB;MAAA,uE;MAAA,0C;QAIyC,gBAA9B,mBAAoB,QAAPB,C;QAAoC,aJvC3C,W;QIUCA,OJt CO,SIscmD,Q;O;KAJ9D,C;IAQA,mC;MAOqB,Q;MAAA,kC;MAAjB,iBAAc,CAAd,yB;QACI,sBAAK,KAAL,E AAc,KAAd,C;;K;IAIR,+B;MAMuD,sBAAQ,4BAAR,C;K;IAEvD,6B;MAIwE,kBAAhB,0B;MAAwB,uB;MAAxB, OJIE7C,W;K;IIOEX,4B;MAQI,gBAAGB,SAAhB,EAA8B,cAAtB,C;K;IAGJ,2C;MAQI,gBAAGB,SAAhB,EAA8B, UAAtB,C;K;IAGJ,2C;MACI,IAAI,IAAK,KAAL,IAAa,CAAjB,C;QAAoB,M;MAEPB,YAAY,YAAY,IAAZ,C;MA CZ,gBAAc,KAAd,EAAqB,UAArB,C;MAEA,aAAU,CAAV,MAAkB,KAAM,OAAXB,M;QACI,iBAAK,CAAL,E AAU,MAAM,CAAN,CAAV,C;;K;IAIR,uC;MACI,OAAO,gBAAkB,IAAIB,O;K;IAGX,iF;MAII,oCAAA,2BAAkB, UAAIB,EAA8B,QAA9B,EAAwC,MAAO,OAA/C,C;MACb,gBAAGB,WAAW,UAAAX,I;MACHB,oCAAA,2BAAk B,iBAAIB,EAAqC,oBAAoB,SAAPB,IAArC,EAAoE,WAAAY,OAAhF,C;MAEb,IAAI,kBAAkB,WAAIB,KAAkC,k BAAkB,MAAIB,CAATC,C;QACI,eAAsB,MAAY,UAAAS,UAAAT,EAAqB,QAArB,C;QACtB,WAAAY,KAAI,QAAJ,

EAAC,iBAAd,C;;QAExB,IAAI,WAAW,WAAAX,IAA0B,qBAAqB,UAAAnD,C;UACI,iBAAC,CAAd,UAAAsB,SAAtB ,U;YACI,YAAAY,oBAAoB,KAApB,IAAZ,IAAyC,OAAO,aAAa,KAAb,IAAP,C;;;UAG7C,mBAAC,YAAAY,CAAZ, IAAd,aAAmC,CAAnC,Y;YACI,YAAAY,oBAAoB,OAApB,IAAZ,IAAyC,OAAO,aAAa,OAAb,IAAP,C;;;K;8GAM zD,qB;MAEGf,gB;K;kGAehF,yB;MAAA,4D;MAAA,4B;QAC8E,OAAK,aAAL,SAAK,C;O;KADnF,C;sGAIA,gC ;MAEI,OAAI,SAAJ,GAEL,SAFJ,GAII,SN63BoB,Q;K;IMz3B5B,mC;MAEI,IAAI,QAAQ,CAAZ,C;QACI,oB;;MA EJ,OAAO,K;K;IAGX,mC;MAEI,IAAI,QAAQ,CAAZ,C;QACI,oB;;MAEJ,OAAO,K;K;IAIX,mC;MAIqD,mB;K;IA ErD,wC;MP1NI,IAAI,EOiOI,YAAAY,CPjOhB,CAAJ,C;QACI,cOgOqB,gC;QP/NrB,MAAM,gCAAYB,OAAQ,WA AjC,C;;K;IOkOd,8C;MAAoE,Y;K;I0D3PV,qC;MAAiC,6B;K;uDAlvF,mB;MACI,qB;MACA,eAAe,e;MACf,OAA O,QAAS,UAAhB,C;QACI,IAAI,OAAA,QAAS,OAAT,EAAMB,OAAAnB,CAAJ,C;UACI,QAAS,S;UACT,OAAO,I ;;;MAGf,OAAO,K;K;yDAGX,oB;MAGoB,Q;MAFhB,qB;MACA,eAAe,K;MACC,0B;MAAhB,OAAGB,cAAhB,C ;QAAgB,yB;QACZ,IAAI,eAAI,OAAJ,CAAJ,C;UAakB,WAAW,I;;MAEjC,OAAO,Q;K;IAKuC,sE;MAAA,qB;Q AAE,OAAM,gBAAN,mB;O;K;4DAFpD,oB;MAEY,Q;MADR,qB;MACA,OAAoC,YAA5B,iEAA4B,EAAU,oDA AV,C;K;IAKU,sE;MAAA,qB;QAAE,QAAO,gBAAP,mB;O;K;4DAFpD,oB;MAEY,Q;MADR,qB;MACA,OAAoC ,YAA5B,iEAA4B,EAAU,oDAAV,C;K;gDAGxY,Y;MACI,qB;MACA,eAAe,IAAK,W;MACpB,OAAO,QAAS,UA AhB,C;QACI,QAAS,O;QACT,QAAS,S;;K;iDAIjB,Y;MAE8B,OAAA,IAAK,U;K;yDAGnC,Y;K;;IC3CgD,+B;MA AiC,oC;MACjF,gBAA8B,C;K;8CAM9B,mB;MAMI,qB;MACA,iBAAI,SAAJ,EAAU,OAAV,C;MACA,OAAO,I; K;mDAGX,2B;MAMc,UACF,M;MANR,oCAAA,4BAAMB,KAAAnB,EAA0B,SAAI1B,C;MAEb,qB;MACA,aAAa, K;MACb,cAAc,K;MACJ,0B;MAAV,OAAU,cAAV,C;QAAU,mB;QACN,kBAAI,eAAJ,EAAI,uBAAJ,WAAc,CA Ad,C;QACA,UAAU,I;;MAEd,OAAO,O;K;0CAGX,Y;MACI,qB;MACA,yBAAY,CAAZ,EAAe,SAAf,C;K;IAKiB, gE;MAAA,qB;QAAE,OAAM,gBAAN,mB;O;K;sDAFvB,oB;MACI,qB;MACA,OAAO,kBAAU,8CAAV,C;K;IAK U,gE;MAAA,qB;QAAE,QAAO,gBAAP,mB;O;K;sDAFvB,oB;MACI,qB;MACA,OAAO,kBAAU,8CAAV,C;K;6C AIX,Y;MAAQd,iD;K;mDAErD,mB;MAAoD,0BAAQ,OAAR,KAAoB,C;K;kDAExE,mB;MACqB,Q;MAAA,6B; MAAjB,iBAAC,CAAd,yB;QACI,IAAI,wBAAI,KAAJ,GAAC,OAAd,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K ;sDAGX,mB;MACI,iBAAC,sBAAd,WAA+B,CAA/B,U;QACI,IAAI,wBAAI,KAAJ,GAAC,OAAd,CAAJ,C;UACI, OAAO,K;;;MAGf,OAAO,E;K;iDAGX,Y;MAA6D,iCAAA,CAAb,C;K;yDAC7D,iB;MAAuE,sDAAiB,KAAjB,C;K ;oDAGvE,8B;MAA4E,uCAAQ,IAAR,EAAc,SAAd,EAAyB,OAazB,C;K;wDAE5E,8B;MAII,eAAe,0BAAa,SAAb ,C;MACf,YAAO,UAAU,SAAV,I;M/DuDX,iBAAC,CAAd,UAAAsB,KAAtB,U;Q+DtDiB,e;QACA,iB;;K;2CAIjB,iB ;MAMI,IAAI,UAAU,IAAd,C;QAAoB,OAAO,I;MAC3B,IAAI,2BAAJ,C;QAAuB,OAAO,K;MAE9B,OAAO,oCA Aa,uBAAC,IAAd,EAAoB,KAApB,C;K;6CAGxB,Y;MAG+B,OAAA,oCAAA,yBAAGB,IAAhB,C;K;IAG5C,kD;M AAA,oB;MACI,eACsB,C;MACTb,cAIqB,E;K;yDAErB,Y;MAAkC,sBAAQ,gB;K;sDAE1C,Y;MAEW,Q;MADP,I AAI,CAAC,cAAL,C;QAAGB,MAAM,6B;MACTb,eAAO,mBAAP,EAAO,2BAAP,O;MACA,OAAO,wBAAI,WA AJ,C;K;wDAGX,Y;MIE5CJ,IAAI,EkE6CU,gBAAQ,EIE7CIB,CAAJ,C;QACI,ckE4CwB,sE;QIE3CxB,MAAM,6B AAsB,OAAQ,WAA9B,C;;MkE6CF,6BAAS,WAAT,C;MACA,eAAQ,W;MACR,cAAO,E;K;;IAOqB,6D;MAHpC, oB;MAGmD,wD;MAG3C,oCAAA,4BAAMB,KAAAnB,EAA0B,WAAyB,KAAAnD,C;MACb,eAAa,K;K;iEAGjB,Y; MAAsC,sBAAQ,C;K;+DAE9C,Y;MAAGC,mB;K;8DAEHc,Y;MACI,IAAI,CAAC,kBAAL,C;QAAoB,MAAM,6B; MAE1B,eAAO,mCAAP,EAAO,YAAP,C;MACA,OAAO,wBAAI,WAAJ,C;K;mEAGX,Y;MAAoC,sBAAQ,CAAR ,I;K;+DAEpC,mB;MACI,wBAAI,YAAJ,EAAW,OAAAX,C;MACA,mC;MACA,cAAO,E;K;+DAGX,mB;MIEIFJ,IA AI,EkEmFU,gBAAQ,EIEnFIB,CAAJ,C;QACI,ckEkFwB,4E;QIEjFxB,MAAM,6BAAsB,OAAQ,WAA9B,C;;MkEk FF,wBAAI,WAAJ,EAAU,OAAV,C;K;;IAIgb,+D;MAAuF,8B;MAAtF,kB;MAA0C,4B;MAC/D,eAAyB,C;MAGr B,oCAAA,2BAAkB,gBAAlB,EAA6B,OAA7B,EAAcS,WAAK,KAA3C,C;MACb,eAAa,UAAU,gBAAV,I;K;wDA GjB,0B;MACI,oCAAA,4BAAMB,KAAAnB,EAA0B,YAA1B,C;MAEb,WAAK,aAAI,mBAAY,KAAZ,IAAJ,EAAuB ,OAAvB,C;MACL,mC;K;wDAGJ,iB;MACI,oCAAA,2BAAkB,KAAIB,EAAyB,YAAzB,C;MAEb,OAAO,wBAAK ,mBAAY,KAAZ,IAAL,C;K;6DAGX,iB;MACI,oCAAA,2BAAkB,KAAIB,EAAyB,YAAzB,C;MAEb,aAAa,WAAK ,kBAAS,mBAAY,KAAZ,IAAT,C;MACIB,mC;MACA,OAAO,M;K;wDAGX,0B;MACI,oCAAA,2BAAkB,KAAIB, EAAyB,YAAzB,C;MAEb,OAAO,WAAK,aAAI,mBAAY,KAAZ,IAAJ,EAAuB,OAAvB,C;K;mGAGO,Y;MAAQ, mB;K;2DAE/B,Y;MAA+C,WAAK,iB;K;;ICxMN,8B;MAAiC,sB;MAwCnF,uBAAoC,I;MA+CpC,yBAA6C,I;K;I AIFR,oD;MAAC,wB;MAGlC,gBAAqB,K;K;iFAHa,Y;MAAA,yB;K;uGAKZ,Y;MAAQ,oB;K;8DAE9B,oB;MAKI, eAAe,IAAK,S;MACpB,gBAAC,Q;MACd,OAAO,Q;K;wDAGX,Y;MAA+B,iEAAc,IAAd,C;K;wDAC/B,Y;MAAK

C,iEAAC,IAAd,C;K;sDACIC,iB;MAA4C,+DAA Y,IAAZ,EAakB,KAAIB,C;K;;IAIB5C,8E;MAAA,wE;MAAsC,2  
CAAK,KAAM,IAAX,EAAGB,KAAM,MAAtB,C;MAAtC,Y;K;IASBJ,+C;MACsE,6B;K;mEACIE,mB;MAAmD,k  
CAAc,OAAAd,C;K;IEAEnD,mB;MAAiD,gCAAY,OAAZ,C;K;;yCAIrD,Y;MACI,YAAQ,Q;K;IAOQ,+F;MAAA,sD  
;MAAS,6B;K;uFACb,mB;MAAwC,MAAM,qCAA8B,8BAA9B,C;K;mFAC9C,Y;MACI,4BAAwB,Q;K;4FAG5B,  
mB;MAAsD,sDAAY,OAAZ,C;K;IAI3C,oH;MAAA,kD;K;4GACH,Y;MAAkC,OAAA,0BAAc,U;K;yGACHd,Y;M  
AAyB,OAAA,0BAAc,OAAO,I;K;2GAC9C,Y;MAAwB,0BAAc,S;K;;sFAL9C,Y;MACI,oBAAoB,oCAAQ,W;MA  
C5B,6G;K;0FAOJ,mB;MACI,qB;MACA,IAAI,+CAAY,OAAZ,CAAJ,C;QACI,4BAAwB,cAAO,OAAP,C;QACxB  
,OAAO,I;;MAEX,OAAO,K;K;oIAGY,Y;MAAQ,OAAA,4BAAwB,K;K;4FAEvD,Y;MAAsC,4BAAwB,iB;K;;0FA  
9B1E,Y;MACI,IAAI,4BAAJ,C;QACI,6F;;MA+BJ,OAAO,mC;K;kDAKf,gB;MAEyB,Q;MADrB,qB;MACqB,OA  
AA,I5EgR2D,QAAQ,W;M4EhRxF,OAAqB,cAArB,C;QAAqB,wB;QAAf,U5EmMsD,U;Q4EnMjD,Y5EgNiD,Y;Q  
4E/MxD,iBAAI,GAJ,EAAS,KAAT,C;;K;IAQc,iG;MAAA,sD;MAAS,oC;K;yFACf,mB;MAAwC,MAAM,qCAA  
8B,gCAA9B,C;K;qFAC9C,Y;MAAuB,4BAAwB,Q;K;8FAE/C,mB;MAAsD,wDAAc,OAAAd,C;K;IAI3C,sH;MAA  
A,kD;K;8GACH,Y;MAAkC,OAAA,0BAAc,U;K;2GACHd,Y;MAAyB,OAAA,0BAAc,OAAO,M;K;6GAC9C,Y;M  
AAwB,0BAAc,S;K;;wFAL9C,Y;MACI,oBAAoB,oCAAQ,W;MAC5B,+G;K;sIAOmB,Y;MAAQ,OAAA,4BAAwB  
,K;K;8FAEvD,Y;MAAsC,4BAAwB,iB;K;;4FAnB1E,Y;MACI,IAAI,8BAAJ,C;QACI,iG;;MAoBJ,OAAO,qC;K;gD  
AGf,e;MACI,qB;MACA,WAAW,YAAQ,W;MACnB,OAAO,IAAK,UAAZ,C;QACI,YAAY,IAAK,O;QACjB,QA  
AQ,KAAM,I;QACd,IAAI,YAAO,CAAP,CAAJ,C;UACI,YAAY,KAAM,M;UACIB,IAAK,S;UACL,OAAO,K;;M  
AGf,OAAO,I;K;kDAIX,Y;K;;IC3I+C,8B;MAAiC,oC;K;0CAEHf,iB;MAMI,IAAI,UAAU,IAAd,C;QAAoB,OAAO  
,I;MAC3B,IAAI,0BAAJ,C;QAAsB,OAAO,K;MAC7B,OAAO,mCAAY,mBAAU,IAAV,EAAGB,KAAhB,C;K;4C  
AGvB,Y;MAG+B,OAAA,mCAAY,2BAAkB,IAAIB,C;K;;ICbT,0B;MAAuD,8B;MAAIC,4B;MACvD,4BAAkC,K;  
K;gCAkBlC,Y;MAEI,qB;MACA,4BAAa,I;MACb,OAAO,I;K;qCAGX,Y;K;iDAGA,uB;K;iFAG8B,Y;MAAQ,OA  
AA,oBAAM,O;K;sCAC5C,iB;MACyC,Q;MAAA,oCAAM,0BAAW,KAAX,CAAN,4D;K;sCACzC,0B;MAIW,IA  
Aa,I;MAHpB,qB;MACA,0BAAW,KAAX,C;MAEoB,gBAAb,qBAAM,KAAN,C;MAAQB,qC;MAA5B,OAAO,CA  
Aa,OIE8BjB,SkE9BI,2D;K;oCAGX,mB;MACI,qB;MACM,oBAAY,MAAK,OAAL,C;MACIB,qC;MACA,OAAO,  
I;K;sCAGX,0B;MACI,qB;MACM,oBAAY,QAAO,mCAAoB,KAApB,CAAP,EAAMC,CAAnC,EAAsC,OAAtC,C;  
MACIB,qC;K;yCAGJ,oB;MACI,qB;MACA,IAAI,QAAS,UAAb,C;QAAwB,OAAO,K;MAE/B,uBAAA,oBpEioDo  
B,QMhrD0C,Y8D+CrD,Q9D/CqD,CNgrD1C,C;MoEhoDpB,qC;MACA,OAAO,I;K;yCAGX,2B;MACI,qB;MACA  
,mCAAoB,KAApB,C;MAEA,IAAI,UAAAS,SAAb,C;QAAmB,OAAO,oBAAO,QAAP,C;MAC1B,IAAI,QAAS,UA  
Ab,C;QAAwB,OAAO,K;MAE3B,IADE,KACF,e;QAAQ,OAAO,oBAAO,QAAP,C;WACf,IAFE,KAEF,O;QAAK,  
uB9D5DqD,Y8D4D7C,Q9D5D6C,CNgrD1C,QoEpnD6B,oBpEonD7B,C;;QoEnnDR,uBAAoC,cAA5B,oBAA4B,E  
AAV,CAAU,EAAP,KAAO,CAAY,Q9D7DE,Y8D6DK,Q9D7DL,C8D6DF,EAA4C,cAAN,oBAAM,EAAY,KA AZ  
,EAAMB,SAAnB,CAA5C,C;;MAG5D,qC;MACA,OAAO,I;K;2CAGX,iB;MACI,qB;MACA,0BAAW,KAAX,C;M  
ACA,qC;MACA,OAAW,UAAS,sBAAb,GACG,oBAAY,MADf,GAGG,oBAAY,QAAO,KAAP,EAAC,CAAd,CAA  
lB,CAAMC,CAAnC,C;K;uCAGR,mB;MAEkB,Q;MADd,qB;MACc,2B;MAAd,mD;QACI,IAAI,4BAAM,KAAN,  
GAAgB,OAAhB,CAAJ,C;UACU,oBAAY,QAAO,KAAP,EAAC,CAAd,C;UACIB,qC;UACA,OAAO,I;;MAGf,OA  
AO,K;K;8CAGX,8B;MACI,qB;MACA,qC;MACM,oBAAY,QAAO,SAAP,EAakB,UAAU,SAAV,IAAIB,C;K;gC  
AGtB,Y;MACI,qB;MACA,uBhChHuC,E;MgCiHvC,qC;K;wCAIJ,mB;MAA+C,OAAM,QAAN,oBAAM,EA AQ,O  
AAR,C;K;4CAErD,mB;MAAmD,OAAM,YAAN,oBAAM,EAAY,OAAZ,C;K;mCAEzD,Y;MAA0B,uBAAc,oBA  
Ad,C;K;0CAE1B,iB;MAGe,UAGL,MAHK,EAMO,M;MAPIB,IAAI,KAAM,OAAN,GAAa,SAAjB,C;QACI,OAA  
O,2D;;MAGc,gBAAxB,eAAK,SAAL,IAAK,gBAAL,yB;MpEuwBL,UAAU,SAAV,EoEvwBsC,KpEuwBtC,EAD+  
F,CAC/F,EADoH,CACpH,EADuI,gBACvI,C;MoErwBI,IAAI,KAAM,OAAN,GAAa,SAAjB,C;QACI,MAAM,SA  
AN,IAAc,6E;;MAGIB,OAAO,K;K;kCAGX,Y;MACI,OAAO,EAAS,MAAM,MAAK,oBAAL,C;K;yCAI1B,Y;MA  
CI,IAAI,yBAAJ,C;QAAGB,MAAM,oC;K;+CAG1B,iB;MACI,oCAAa,kCAAYB,SAAZB,C;MADoB,Y;K;wDAIrC,  
iB;MACI,oCAAa,mCAA0B,SAAI1B,C;MAD6B,Y;K;;IAI9C,+B;MAAA,mD;MAG8B,sBhCRa,EgCQb,C;MAH9  
B,Y;K;IAKA,kD;MAAA,mD;MAIkD,sBhCdP,EgCcO,C;MAJID,Y;K;IAMA,2C;MAAA,mD;MAGqD,sB9DLA,Y8  
DKR,Q9DLQ,C8DKb,C;MAHrD,Y;K;ICrBJ,0C;MACI,IAAI,6BAAJ,C;QACU,KAAY,MAAK,UAAAL,C;;QAEIB,  
UAAU,KA AV,EA AwC,CAAxC,EAAiD,cAAN,KAAM,CAAjD,EAA4D,eAAW,UAA X,CAA5D,C;;K;IAMiB,kD;  
MAAA,uB;QAAGB,OAAA,kBAAW,SAAQ,CAAR,EA AW,CAAX,C;O;K;IAFpD,4C;MACI,IAAI,6BAAJ,C;QAC

I,iBAAiB,gC;QACX,KAAy,MAAK,UAAAL,C;;QAEiB,UAAU,KAAV,EAAwC,CAAxC,EAAiD,cAAN,KAAM,C  
AAjD,EAA4D,UAA5D,C;;K;IAIR,gE;MACI,IAAI,aAAY,UAAU,CAAV,IAAZ,CAAJ,C;QACI,UAAU,KAAV,EA  
AwC,SAAxC,EAAMd,UAAU,CAAV,IAAnD,EAAGe,UAAhE,C;;K;IAMiB,gC;MAAgB,OAAE,iBAAF,CAAE,E  
AAU,CAAV,C;K;IAF3C,0B;MACI,IAAI,6BAAJ,C;QACI,iBAAiB,gB;QACX,KAAy,MAAK,UAAAL,C;;QAEiB,  
UAAU,KAAV,EAAwC,CAAxC,EAAiD,cAAN,KAAM,CAAjD,EAA4D,cAA5D,C;;K;;IAaa,kD;MAAoB,QAAC,I  
AAM,CAAP,KAAa,IAAM,CAAnB,K;K;IARzC,uC;MACI,sC;QAAiC,OAAjC,yB;;MACA,4BAA4B,K;MAE5B,Y  
AAy,E;MAGZ,iBAAc,CAAd,UAAsB,GAAtB,U;QAAiC,KAAy,MAAK,KAAL,C;MAC7C,iBAAiB,kC;MACX,K  
AAy,MAAK,UAAAL,C;MACiB,mBAAc,CAAd,YAAsB,KAAM,OAA5B,Y;QACI,QAAQ,MAAM,UAAQ,CAAR,I  
AAN,C;QACR,QAAQ,MAAM,OAAN,C;QACR,IAAI,CAAC,IAAM,CAAP,OAAc,IAAM,CAApB,KAA0B,KAA  
K,CAAnC,C;UAAsC,OAAO,K;;MAEjD,4BAA4B,I;MAC5B,OAAO,I;K;IAIX,2D;MACI,aAAa,gBAAMb,KAAM  
,OAAzB,O;MACb,aAAa,YAAU,KAAV,EAAiB,MAAjB,EAAYB,KAAzB,EAAGC,YAAhC,EAA8C,UAA9C,C;M  
ACb,IAAI,WAAW,KAAf,C;QACI,aAAU,KAAV,OAAiB,YAAjB,M;UAA+B,MAAM,CAAN,IAAW,OAAO,CAA  
P,C;;K;IAIID,4D;MAEI,IAAI,UAAAS,GAAb,C;QACI,OAAO,K;;MAGX,aAAa,CAAC,QAAQ,GAAR,IAAD,IAAg  
B,CAAhB,I;MACb,WAAW,YAAU,KAAV,EAAiB,MAAjB,EAAYB,KAAzB,EAAGC,MAAhC,EAAwC,UAAxC,  
C;MACX,YAAy,YAAU,KAAV,EAAiB,MAAjB,EAAYB,SAAS,CAAT,IAAZB,EAAGC,GAAR,C,EAA0C,UAA1C,  
C;MAEZ,aAAiB,SAAS,MAAb,GAAqB,KAArB,GAAgC,M;MAG7C,gBAAGB,K;MAChB,iBAAiB,SAAS,CAAT,  
I;MACjB,aAAU,KAAV,OAAiB,GAAjB,M;QAEQ,iBAAa,MAAb,IAAuB,cAAc,GAAR,C;UACI,gBAAGB,KAA  
K,SAAL,C;UAChB,iBAAiB,MAAM,UAAAN,C;UAEjB,IAAI,UAAW,SAAQ,SAAR,EAAMb,UAAAnB,CAAX,IAA  
6C,CAAjD,C;YACI,OAAO,CAAP,IAAY,S;YACZ,6B;;YAEA,OAAO,CAAP,IAAY,U;YACZ,+B;;eAGR,iBAAa,  
MAAb,C;UACI,OAAO,CAAP,IAAY,KAAK,SAAL,C;UACZ,6B;;UAGA,OAAO,CAAP,IAAY,MAAM,UAAAN,C;  
UACZ,+B;;MAMZ,OAAO,M;K;ICrGX,4C;MAMoB,UACM,M;MAHtB,IAAI,iBAAJ,C;QAAkB,OAAO,C;MACz  
B,aAAa,C;MACb,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAEQ,oB;UAAmB,U;;UACnB,I5BFiC,MAAa  
,Y4BEEnC,O5BFmC,C4BE9C,C;YAAwD,iCAAhC,OAAGC,C;iBAExD,uC;YAAmC,2BAAR,OAAQ,C;eACnC,wC  
;YAAmC,2BAAR,OAAQ,C;eACnC,sC;YAAmC,2BAAR,OAAQ,C;eACnC,uC;YAAmC,2BAAR,OAAQ,C;;YAE  
A,kBAAR,OAAQ,C;;QATvC,wB;QAYA,SAAS,MAAK,MAAL,QAAC,WAAAd,I;;MAEb,OAAO,M;K;;ICTP,uC;  
MAAA,2C;K;2DACI,0B;MAA2D,sBAAU,MAAV,C;K;gEAE3D,iB;MAA6C,Q;MAAA,wEAAqB,C;K;;IAHtE,m  
D;MAAA,kD;QAAA,iC;;MAAA,2C;K;;MC0BA,iC;MAKA,8B;MA6CA,0BAAMe,I;;IAzEnE,kC;MAAA,oB;MA  
A+B,8C;K;2CAE3B,mB;MAAyD,MAAM,qCAA8B,iCAA9B,C;K;uCAC/D,Y;MACI,WAAa,Q;K;uDAGjB,mB;M  
AAgE,OAAA,WAAa,uBAAc,OAAd,C;K;0CAE7E,Y;MAAwE,OAAA,iCAAY,W;K;qDAEpF,mB;MACI,IAAI,iB  
AAS,OAAT,CAAJ,C;QACI,WAAa,cAAO,OAAQ,IAAf,C;QACb,OAAO,I;;MAEX,OAAO,K;K;wFAGY,Y;MAA  
Q,OAAA,WAAa,K;K;;8BA6ChD,Y;MACI,0BAAY,Q;K;0CAIhB,e;MAAMd,OAAA,0BAAY,gBAAS,GAAT,C;K  
;4CAE/D,iB;MAAMe,gBAAZ,0B;MAAY,c;;QnEinDnD,Q;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,aAAO,K  
;UAAP,e;;QACrB,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAAM,ImEjnDmD,uBAAS,gBnEinD9C,OmEjnDw  
D,MAAV,QnEinD5D,C;YAAwB,aAAO,I;YAAP,e;;QAC9C,aAAO,K;;MmElnDgD,iB;K;kFAInD,Y;MACI,IAAI,  
+BAAJ,C;QACI,0BAAW,qB;;MAEf,OAAO,sC;K;uCAGf,Y;MAAgF,iC;K;kCAEhF,e;MAA+C,OAAA,0BAAY,  
WAAI,GAAJ,C;K;oCAE3D,sB;MAAgD,OAAA,0BAAY,aAAI,GAAJ,EAAS,KAAT,C;K;qCAE5D,e;MAAyC,OA  
AA,0BAAY,cAAO,GAAP,C;K;+EAEvB,Y;MAAQ,OAAA,0BAAY,K;K;;IA5DID,0C;MAAA,iD;MAAuD,8B;MA  
vC3D,mB;MAwCQ,8BAAMb,W;MACnB,2BAAGB,WAAy,S;MAFhC,Y;K;IAKA,+B;MAAA,iD;MAGuB,aAAK  
,kEAAL,Q;MAHvB,Y;K;IAKA,4D;MAAA,iD;MAQ8D,qB;MzEpC9D,IAAI,EyEsCQ,mBAAMb,CzEtC3B,CAAJ,  
C;QACI,cyEqCgC,+C;QzEpChC,MAAM,gCAAYB,OAAQ,WAAjC,C;;MAFV,IAAI,EyEuCQ,cAAc,CzEvCtB,CA  
AJ,C;QACI,gByEsC2B,yC;QzErC3B,MAAM,gCAAYB,SAAQ,WAAjC,C;;MyE0BV,Y;K;IAcA,gD;MAAA,iD;M  
AA2C,eAAK,eAAL,EAAsB,GAAtB,Q;MAA3C,Y;K;IAGA,yC;MAAA,iD;MAG8C,qB;MAC1C,KAAK,gBAAO,  
QAAP,C;MAJT,Y;K;IAqCJ,4B;MAK8E,gBAAnE,aAAmB,gEAAAnB,C;MAA2E,wB;MAAIF,OtEvCo,S;K;;MuEjE  
P,uB;;kCAyCA,mB;MACI,UAAU,gBAAI,aAAI,OAAJ,EAAa,IAAb,C;MACd,OAAO,W;K;8BAGX,Y;MACI,gBA  
AI,Q;K;uCAOR,mB;MAA6D,OAAA,gBAAI,mBAAY,OAAZ,C;K;gCAEjE,Y;MAAyC,OAAA,gBAAI,U;K;iCAE  
7C,Y;MAAQD,OAAA,gBAAI,KAAK,W;K;qCAE9D,mB;MAAKD,OAAA,gBAAI,cAAO,OAAP,CAAJ,Q;K;+EA  
EpB,Y;MAAQ,OAAA,gBAAI,K;K;;IA5D1C,6B;MAAA,iD;MAGoB,8B;MAZxB,mB;MAAQ,oBAAM,gB;MAJV,  
Y;K;IAOA,yC;MAAA,iD;MAG2C,8B;MAnB/C,mB;MAoBQ,oBAAM,eAAGB,QAAS,KAAzB,C;MACN,qBAAO,



QAAP,C;MALJ,Y;K;IAQA,4D;MAAA,iD;MAQ2D,8B;MAhC/D,mB;MAiCQ,oBAAM,eAAgB,eAAhB,EAAiC,U  
AAjC,C;MATV,Y;K;IAYA,gD;MAAA,iD;MAA2C,eAAK,eAAL,EAA5B,GAATB,Q;MAA3C,Y;K;IAEA,oC;MA  
AA,iD;MAM0C,8B;MA5C9C,mB;MA6CQ,oBAAW,G;MAPf,Y;K;IAmCJ,+B;MAKuC,gBAA5B,eAAQ,eAAR,C;  
MAAoC,6B;MAA3C,OvENO,S;K;IwEzD6B,uC;MAAC,kC;MAErC,oBAAkC,kB;MACIC,sBAAYB,C;K;2EAHY,  
Y;MAAA,8B;K;2FAGrC,Y;MAAA,0B;K,OAAA,gB;MAAA,0B;K;gDAGA,sB;MACI,eAAe,aAAS,qBAAY,GAA  
Z,C;MACxB,mBAAmB,6BAAsB,QAATB,C;MACnB,IAAI,oBAAJ,C;QAEI,kBAAW,QAAX,IAAuB,mCAAY,GA  
AZ,EAAiB,KAAjB,C;;QAEvB,IAAI,6BAAJ,C;UAEI,YAA+B,Y;UAC/B,IAAI,aAAS,gBAAO,KAAM,IAAb,EAA  
kB,GAAIB,CAAb,C;YACI,OAAO,KAAM,gBAAS,KAAT,C;;YAEb,kBAAW,QAAX,IAAuB,CAAQ,KAAR,EAA  
e,mCAAY,GAAZ,EAAiB,KAAjB,CAAf,C;YACvB,6B;YACA,OAAO,I;;;UAIX,YAAuC,Y;UACvC,cAAkB,wBA  
AN,KAAM,EAAiB,GAAjB,C;UACIB,IAAI,eAAJ,C;YACI,OAAO,OAAM,gBAAS,KAAT,C;;UAEX,KAAY,MA  
AK,mCAAY,GAAZ,EAAiB,KAAjB,CAAL,C;;;MAG1B,6B;MAEA,OAAO,I;K;iDAGX,e;MAEuB,Q;MADnB,eA  
Ae,aAAS,qBAAY,GAAZ,C;MACL,oCAAsB,QAATB,C;MAAA,iB;QAAmC,OAAO,I;;MAA7D,mBAAmB,I;MAC  
nB,IAAI,6BAAJ,C;QACI,YAAgC,Y;QACHC,IAAI,aAAS,gBAAO,KAAM,IAAb,EAAkB,GAAIB,CAAb,C;U9Bz  
DR,O8B0D6B,iB9B1DvB,C8B0DmC,Q9B1DnC,C;U8B2DM,6B;UACA,OAAO,KAAM,M;;UAEb,OAAO,I;;;QA  
GX,YAAuC,Y;QACvC,8BAAc,KAAd,iB;UACI,cAAY,MAAM,KAAN,C;UACZ,IAAI,aAAS,gBAAO,GAAP,EA  
AY,OAAM,IAAIB,CAAb,C;YACI,IAAI,KAAM,OAAN,KAAC,CAAIB,C;cACU,KAAN,UAA2B,C;c9BtE/C,O8B  
wEqC,iB9BxE/B,C8BwE2C,Q9BxE3C,C;;c8B2EoB,KAAY,QAAO,KAAP,EAAc,CAAd,C;;YAEtB,6B;YAEA,OA  
AO,OAAM,M;;;MAIzB,OAAO,I;K;OCAGX,Y;MACI,oBAAa,kB;MACb,YAAO,C;K;mDAGX,e;MAAYC,uBAA  
S,GAAT,S;K;8CAEzC,e;MAA+B,Q;MAAA,+BAAS,GAAT,8B;K;+CAE/B,e;MACuB,Q;MAAA,oCAAsB,aAAS,  
qBAAY,GAAZ,CAA/B,C;MAAA,iB;QAAoD,OAAO,I;;MAA9E,mBAAmB,I;MACnB,IAAI,6BAAJ,C;QACI,YA  
AgC,Y;QACHC,IAAI,aAAS,gBAAO,KAAM,IAAb,EAAkB,GAAIB,CAAb,C;UACI,OAAO,K;;UAEP,OAAO,I;;;Q  
AGX,YAAuC,Y;QACvC,OAAa,wBAAN,KAAM,EAAiB,GAAjB,C;;K;uDAlrB,0B;MACI,sB;;Q1FsoCY,Q;QAAh  
B,iD;UAAgB,cAAhB,e;UAA5B,I0FtoCK,aAAS,gB1FsoCA,O0FtoCa,IAAb,M1FsoCd,C;YAAwB,qBAAO,O;YAA  
P,uB;;;QAC9C,qBAAO,I;;M0FvoCH,yB;K;IAIO,8E;MAAA,wD;MACH,aAAY,E;MAEZ,YAA0B,MAAa,MAAK  
,qCAAL,C;MACvC,gBAAe,E;MAEf,oBAA4B,I;MAC5B,eAAc,K;MACd,iBAAgB,E;MACHB,iBAAqC,I;K;yEAE  
rC,Y;MACI,IAAI,6BAAwB,YAA5B,C;QACI,gBAAqB,iBAAqD,O;QAC1E,IAAI,4DAAC,SAIIB,C;UACI,OAAO  
,C;;MAGf,IAAI,yDAAa,SAAK,OAAtB,C;QACI,oBAAe,2CAAW,UAAK,aAAL,CAAX,C;QACf,eAAU,iC;QACV  
,iBAAy,C;QACZ,OAAO,C;;QAEp,oBAAe,I;QACf,OAAO,C;;K;mEAlf,Y;MACI,IAAI,eAAS,EAAb,C;QACI,aA  
AQ,oB;MACZ,OAAO,eAAS,C;K;gEAGpB,Y;MAEoB,Q;MADhB,IAAI,CAAC,cAAL,C;QAAGB,MAAM,6B;MA  
CN,IAAI,YAAJ,C;QACZ,yBAAqD,cAArD,C;;QAEa,OAAb,iB;;MAHJ,oB;MAKA,iBAAiB,S;MACjB,aAAQ,E;M  
ACR,OAAO,S;K;KEAGX,Y;M3E/CR,I2EgDyB,c3EhDrB,QA AJ,C;QACI,cAhByB,0B;QAiBzB,MAAM,6BAAsB,  
OAAQ,WAA9B,C;;M2E+CE,6BAAyB,cAAO,6BAAy,IAAnB,C;MACzB,iBAAy,I;MAEZ,uC;K;;6CatDZ,Y;MA  
EI,2D;K;4DAyDJ,oB;MACI,mBAAmB,kBAAW,QAAX,C;MACnB,OA AW,iBAAiB,SAArB,GAAGC,IAAhC,GA  
A0C,Y;K;;;wCCtKrD,Y;MACI,aAAR,MAAM,OAAe,CAAP,IAAO,C;MAEb,OAAO,KAAP,IAAGB,C;M/BXpB,  
O+BYqB,M/BZf,C+BYuB,K/BZvB,C;M+BaF,OAAO,M;K;;ICNuB,qC;MAAC,kC;MAEnC,oBAAkC,kB;MACIC,  
sBAAYB,C;K;yEAHU,Y;MAAA,8B;K;yFAGnC,Y;MAAA,0B;K,OAAA,gB;MAAA,0B;K;iDAWA,e;MACI,IAAI,  
0BAAJ,C;QAAoB,OAAO,K;MAC3B,OAAO,kBAAW,GAAX,MAAoB,S;K;4CAG/B,e;MACI,IAAI,0BAAJ,C;QA  
AoB,OAAO,I;MAC3B,YAAy,kBAAW,GAAX,C;MACZ,OA AW,UAAU,SAArB,GAAGC,KAAhC,GAA2D,I;K;8  
CAI/D,sB;M7EVA,IAAI,E6EWQ,uB7EXR,CAAJ,C;QACI,cAda,qB;QAeb,MAAM,gCAAYB,OAAQ,WAAjC,C;;  
M6EUN,eAAe,kBAAW,GAAX,C;MACf,kBAAW,GAAX,IAAkB,K;MAEIB,IAAI,aAAa,SAAJB,C;QACI,6B;QA  
EA,OAAO,I;;QAGP,OAAO,Q;;K;+CAIf,e;MACI,IAAI,0BAAJ,C;QAAoB,OAAO,I;MAC3B,YAAy,kBAAW,GA  
AX,C;MACZ,IAAI,UAAU,SAAd,C;QhCnDJ,OgCoDyB,iBhCpDnB,CgCoD+B,GhCpD/B,C;QgCqDE,6B;QAEA,  
OAAO,K;;QAGP,OAAO,I;;K;wCAKf,Y;MACI,oBAAa,kB;MACb,YAAO,C;K;IAKA,0E;MAAA,oD;MACH,cAA  
kC,MAAa,MAAK,mCAAL,C;MAC/C,kBAA4B,qBAAL,WAAK,C;MAC5B,iBAA+B,I;K;iEAE/B,Y;MAAKC,OA  
AA,eAAS,U;K;8DAE3C,Y;MAIuB,gB;MAHnB,UAAU,eAAS,O;MACnB,iBAAU,G;MAES,+E;MAAnB,OAAO,i  
D;K;gEAGX,Y;MAEKc,UAA9B,M;MAAA,oC;MAA8B,YAAa,c;M7EchD,uB;MAEP,IAfoB,KAehB,QA AJ,C;QA  
CI,cAhByB,0B;QAiBzB,MAAM,6BAAsB,OAAQ,WAA9B,C;;QAEN,sBAnBgB,K;;M6Ede,oBAAO,sFAAP,C;K;;  
2CAjBnC,Y;MACI,yD;K;IAqBkD,0F;MAAA,8B;MAAA,oD;K;kHAC9B,Y;MAAQ,uB;K;oHACN,Y;MAAQ,6CA

AuB,gBAAvB,C;K;2EAE9B,oB;MAAwC,OAAA,2BAAuB,aAAI,gBAAJ,EAAS,QAAT,C;K;qEAE/D,Y;MAA+B,  
OAAA,mCAAY,uBAAc,IAAd,C;K;qEAC3C,Y;MAAKC,OAAA,mCAAY,uBAAc,IAAd,C;K;mEAC9C,iB;MAA4  
C,OAAA,mCAAY,qBAAY,IAAZ,EAakB,KAAIB,C;K;;gDAR5D,e;MAAsD,iE;K;;;MCItd,sBAOsC,I;MA6CtC,y  
B;MAOA,4BAakC,K;;IArIE,sD;MAZpC,oB;MAYyD,0CAAqC,GAArC,EAA0C,KAA1C,C;MACrD,oBAAuC,I;  
MACvC,oBAAuC,I;K;wDAEvC,oB;MACI,WAAmB,iB;MACnB,OAAa,mEAAS,QAAT,C;K;;IAIrB,wC;MAAA,o  
B;MAA+B,8C;K;IAE3B,sD;MAAA,oB;MACI,cACsC,I;MAEtC,cACsC,I;MAG1C,cAAO,iC;K;6DAIX,Y;MACI,O  
AAO,gBAAS,I;K;0DAGpB,Y;MAEI,IAAI,CAAC,cAAL,C;QAAgB,MAAM,6B;MAEtB,cAAc,0B;MACd,cAAO,  
O;MACa,gBAAb,OAAQ,a;;MAAf,c3E0DS,S2E1DoB,KAAO,iC3E0DzC,GAAqB,SAArB,GAA+B,I;M2EzD1B,O  
AAO,O;K;4DAGX,Y;M9EwBR,IAAI,E8EvBc,eAAQ,I9EuBtB,CAAJ,C;QACI,cAdW,e;QAeX,MAAM,6BAAsB,  
OAAQ,WAA9B,C;;M8ExBE,WAAc,iB;MAGP,oCAAP,0BAAO,C;MACP,gCAAI,cAAO,0BAAO,IAAd,C;MAEJ,  
cAAO,I;K;;iDAIf,mB;MAAyD,MAAM,qCAA8B,iCAA9B,C;K;6CAC/D,Y;MACI,WAAmB,Q;K;6DAGvB,mB;M  
AAgE,OAAA,WAAmB,uBAAc,OAAd,C;K;gDAEnF,Y;MAAwE,qD;K;2DAExE,mB;MACI,qB;MACA,IAAI,iBA  
AS,OAAT,CAAJ,C;QACI,WAAmB,cAAO,OAAQ,IAAf,C;QACnB,OAAO,I;;MAEX,OAAO,K;K;8FAGY,Y;MA  
AQ,OAAA,WAAmB,K;K;sDAEID,Y;MAAsC,WAAmB,iB;K;;iDaa7D,qB;M9ErBA,IAAI,E8E0BM,0BAAQ,IAA  
R,IAAgB,0BAAQ,I9E1B9B,CAAJ,C;QACI,cAdW,e;QAeX,MAAM,6BAAsB,OAAQ,WAA9B,C;;M8E0BN,YAA  
Y,mB;MACZ,IAAI,SAAS,IAAb,C;QACI,sBAAO,S;QACP,yBAAO,S;QACP,yBAAO,S;;QAGK,YAAa,KAAM,a;  
Q9ElBhC,uB;QAeP,IAfoB,KAehB,QA AJ,C;UACI,gBAhByB,0B;UAIbZB,MAAM,6BAAsB,SAAQ,WAA9B,C;;U  
AEN,sBAnBgB,K;;Q8EkBZ,+B;QAEA,yBAAO,K;QACP,yBAAO,K;QAEP,qBAaA,S;QACb,qBAaA,S;;K;+CAIr  
B,qB;MAIL,IAAI,SAAK,aAAL,KAAC,SAAB,C;QAEL,sBAAO,I;;QAEP,IAAI,wBAAS,SAAb,C;UAEL,sBAAO,sB  
;;QAEX,qDAAc,sB;QACd,qDAAc,sB;;MAEIB,yBAAO,I;MACP,yBAAO,I;K;oCA8CX,Y;MAEI,qB;MACA,4BA  
Aa,I;MACb,OAAO,I;K;oCAGX,Y;MACI,qB;MACA,kBAAI,Q;MACJ,sBAAO,I;K;gDASX,e;MAAmD,OAAA,kB  
AAI,mBAAy,GAAZ,C;K;kDAEvD,iB;MACiC,Q;MAAA,0B;MAAA,iB;QAAQ,OAAO,K;;MAA5C,WAA6B,I;;Q  
AEzB,IAAI,OAAA,IAAK,MAAL,EAAC,KAAd,CAAJ,C;UACI,OAAO,I;;QAEX,OAAO,cAAA,IAAK,aAAL,C;;  
MACF,iBAAS,mBAAT,C;MACT,OAAO,K;K;6CAIX,Y;MAAoF,uC;K;wCAEpF,e;MAAmD,Q;MAAJ,QAAL,OA  
AJ,kBAAI,WAAI,GAAJ,CAAJ,6B;K;0CAE/C,sB;MACI,qB;MAEA,UAAU,kBAAI,WAAI,GAAJ,C;MACd,IAAI,  
OAAO,IAAX,C;QACI,eAAe,mCAAW,GAAX,EAAGB,KAAhB,C;QACf,kBAAI,aAAI,GAAJ,EAAS,QAAT,C;QA  
CK,wBAAT,QAAS,C;QACT,OAAO,I;;QAEP,OAAO,GAAL,gBAAS,KAAT,C;;K;2CAInB,e;MACI,qB;MAEA,Y  
AAY,kBAAI,cAAO,GAAP,C;MACHB,IAAI,SAAS,IAAb,C;QACU,sBAAN,KAAM,C;QACN,OAAO,KAAM,M;;  
MAEjB,OAAO,I;K;qFAGmB,Y;MAAQ,OAAA,kBAAI,K;K;6CAE1C,Y;MACI,IAAI,yBAAJ,C;QAAgB,MAAM,  
oC;K;;IANg1B,mC;MAAA,uD;MAGuB,qB;MA9J3B,yB;MA+JQ,sBAAM,gB;MAJv,Y;K;IAOA,iD;MAAA,uD;  
MAAoD,qB;MAIKxD,yB;MAoKc,Q;MAAN,sBAAM,+D;MAFV,Y;K;IAKA,kE;MAAA,uD;MAQ8D,eAAM,eAA  
N,EA AuB,UAAvB,Q;MA/KIE,yB;MAGLQ,sBAAM,gB;MATV,Y;K;IAYA,sD;MAAA,uD;MAA2C,qBAAK,eAA  
L,EAAsB,GAAtB,Q;MAA3C,Y;K;IAEA,+C;MAAA,uD;MAG2C,qB;MAxL/C,yB;MAyLQ,sBAAM,gB;MACN,K  
AAK,gBAAO,QAAP,C;MALT,Y;K;IA6EJ,kC;MAKwD,gBAA7C,qBAAYB,eAAzB,C;MAAqD,wB;MAA5D,O3E  
jMO,S;K;;oC4EvCP,Y;MAEK,Q;MAA8B,CAA9B,2EAA8B,S;MAC/B,OAAO,I;K;6CAGX,Y;MAA+C,gBAALi  
B;K;;IAhCnD,wC;MAAA,uD;MAAmD,eAAM,GAAN,Q;MAPvD,yB;MAOI,Y;K;IAEA,qC;MAAA,uD;MAGuB,e  
AAM,oBAAN,Q;MAZ3B,yB;MASI,Y;K;IAKA,+C;MAAA,uD;MAG8C,eAAM,oBAAN,Q;MAjBID,yB;MAkBQ,  
qBAAO,QAAP,C;MAJJ,Y;K;IAOA,kE;MAAA,uD;MAQ8D,eAAM,qBAAsB,eAAtB,EAAuC,UAAvC,CAAN,Q;  
MA7BIE,yB;MAqBI,Y;K;IAUA,sD;MAAA,uD;MAA2C,qBAAK,eAAL,EAAsB,GAAtB,Q;MAA3C,Y;K;IAGBJ,q  
C;MAKMD,gBAAXC,mBAAc,qBAAd,C;MAAgD,6B;MAAvD,O5EoBO,S;K;;;kF6EzEX,uB;MAQI,OAAO,O;K;I  
CXX,sB;K;mCACI,Y;MACI,mBAAM,IAAN,C;K;2CAGJ,mB;MACI,mBAAM,OAAN,C;MACA,c;K;iCAKJ,Y;K;  
;IAKuB,oC;MAA8B,qB;MAA7B,gC;K;2CACxB,mB;MAEI,oBA+DyC,OA/Dd,OA+Dc,C;MA9DzC,iBAAa,OAA  
M,aAAN,C;K;;IAIrB,8B;MAEoC,qB;K;iDACHC,mB;MACI,OAAQ,KAAI,OAAJ,C;K;mDAGZ,mB;MACI,OAAQ  
,KAAI,OAAJ,C;K;2CAGZ,Y;MACI,OAAQ,KAAI,EA AJ,C;K;;IAIhB,0B;MAEqC,qB;MACjC,cAAa,E;K;6CAEb,  
mB;MACI,eAoCyC,OApCxB,OAoCwB,C;K;qCAjC7C,Y;MACI,cAAS,E;K;;IAIjB,sC;MAE4C,yB;K;yDACxC,m  
B;MACI,QAwByC,OAxB1B,OAxB0B,C;MAvBzC,QAAQ,CpEqJoF,aoErJhE,IpEqJgE,EoErJ1D,CpEqJ0D,C;Mo  
EpJ5F,IAAI,KAAK,CAAT,C;QACI,4BAAU,CpEwL0E,WoExL9D,CpEwL8D,EoExL3D,CpEwL2D,C;QoEvLpF,  
Y;QACA,IAAI,CpEmLiE,WoEnLrD,IAAI,CAAJ,IpEmLqD,C;;MoEjLzE,4BAAU,C;K;iDAGd,Y;MACI,OAAQ,K

AAI,WAAJ,C;MACR,cAAS,E;K;;;IAWjB,yB;MACiD,cAAa,KAAb,C;K;IAEjD,mB;MAEI,MAAO,U;K;IAGX,4B  
;MAEI,MAAO,iBAAQ,OAAR,C;K;IAGX,wB;MAEI,MAAO,eAAM,OAAN,C;K;IAGX,kB;MACqC,MAAM,qCA  
A8B,sCAA9B,C;K;IAE3C,wB;MAC4C,MAAM,qCAA8B,4CAA9B,C;K;ICIGID,mD;MACI,0B;MASA,gBAA2B,  
a;K;2FAFvB,Y;MAAQ,OAAA,eAAS,Q;K;oDAIrb,kB;MACI,UAAU,IAAK,S;MAEX,YAAQ,2CAAR,C;QACI,g  
BAAc,MAAO,M;WAEzB,YAAQ,yBAAR,C;QACI,gBAAc,yC;QACd,eAAS,oBAAW,MAAX,C;;QAEI,MAAM,6  
BAAsB,iBAAtB,C;K;4CAItB,Y;MAOW,Q;MALP,IAAI,kBAAW,2CAAf,C;QACI,gBAAS,yB;QACT,OAAO,yB;;  
MAEX,aAAa,IAAK,S;MAEd,eAAW,yCAAX,C;QAAsB,gC;WACTb,0C;QAA4B,MAAM,MAAO,U;;QACjC,a;M  
AHZ,W;K;;IA7BJ,gD;MAAA,0D;MACyD,6BAAK,QAAL,EAAe,2CAAf,C;MADzD,Y;K;;;;ICRA,2C;MAAA,+D  
;MAAuB,iC;MAF3B,iC;MAEI,Y;K;IACA,sD;MAAA,+D;MAAuC,6BAAM,OAAN,Q;MAH3C,iC;MAGI,Y;K;IA  
CA,6D;MAAA,+D;MAAmD,kCAAM,OAAN,EAAe,KAAf,C;MAJvD,iC;MAII,Y;K;IACA,oD;MAAA,+D;MAAi  
C,6BAAM,KAAQ,Q;MALrC,iC;MAKI,Y;K;I1C4CJ,yE;MASI,sC;MAAA,4C;K;IATJ,iGAWY,Y;MAAQ,2B;KA  
XpB,E;IAAA,0DAaQ,kB;MACI,wBAAW,MAAX,C;K;IAdZ,sF;I2C5C2E,0C;M5CkKhE,Q;MADP,e4ChKA,M5C  
gKA,C;MACO,Q4CjKP,M5CiKO,+D;M4ChKX,W;K;;+FCuHA,gB;MACI,aAAa,IAAb,MAAa,E;MACb,KAAK,M  
AAL,C;MACA,OAAO,M;K;wFC3HX,yB;MAAA,uD;MAAA,wC;QAWqG,OAAK,cAAL,SAAK,EAAiB,IAAjB,E  
AAuB,IAAvB,C;O;KAX1G,C;wFAaA,yB;MAAA,uD;MAAA,wC;QAWoG,OAAK,cAAL,SAAK,EAAiB,IAAjB,E  
AAuB,IAAvB,C;O;KAXzG,C;8ECbA,yB;MAAA,6C;MAAA,sC;QAOyD,OAAK,SAAL,SAAK,EAAy,QAAZ,C;  
O;KAP9D,C;8EASA,yB;MAAA,6C;MAAA,wC;QAWkE,OAAK,SAAL,SAAK,EAAa,UAAb,S;O;KAXvE,C;oFA  
aA,yB;MAAA,mD;MAAA,wC;QAWqE,OAAK,YAAL,SAAK,EAAgB,UAAhB,S;O;KAX1E,C;kFCZI,yB;MAAA  
iD;MAAA,4B;QAAe,OAAK,WAAL,SAAK,C;O;KAApB,C;wFAyA,yB;MAAA,uD;MAAA,4B;QAAe,OAAK,c  
AAL,SAAK,C;O;KAApB,C;IC5BJ,gC;MAAoE,gCAAqB,OAArB,C;K;IAEIC,uC;MAAC,wB;K;iDAC/B,iB;MACI  
eAAQ,KAAAR,C;K;8CAGJ,Y;MAAyC,iCAAuB,cAAvB,M;K;;ICCO,6C;MAAA,8B;MAAS,uB;K;8FACIC,Y;MA  
AQ,OAAA,gBAAY,O;K;mDAE3C,iB;MACI,IADoC,KACpC,IAAG,CAAH,IADoC,KACpC,IAAM,sBAAN,C;Q  
AD8B,OACX,gBAAY,MAAK,KAAL,C;;QACvB,MAAM,8BAA0B,WAAQ,KAAAR,6BAAMc,sBAAnC,MAA1B,  
C;K;;IARtB,8B;MAGoD,4C;K;wECFpD,yB;MAAA,uC;MAAA,4B;QAOsC,MAAL,SAAK,C;O;KAPtC,C;kFASA  
yB;MAAA,iD;MAAA,kC;QAWuD,OAAK,WAAL,SAAK,EAAc,IAAd,C;O;KAX5D,C;+ECfA,qB;MAI8C,gB;K;i  
FAE9C,qB;MAIsE,OAAK,S;K;kFAE3E,qB;MAMyE,gB;K;IAEzE,6B;MAiBa,UAPF,M;MAFP,QAAc,S;MAGV,c  
AAK,UAAAL,U;QACI,mBAAK,UAAAL,G;;QACJ,IjDzBqC,MAAa,YiDyBvC,CjDzBuC,CiDyBiD,C;UAC6B,8BAA  
zB,CAAYB,C;;UAGN,UAAIB,uDAakB,Y;;MAP3B,a;K;IC9BJ,2B;MAEI,MAAM,yBAAqB,OAArB,C;K;IAGV,s  
B;MAEI,MAAM,uBAAMb,cAAAnB,C;K;IAGV,2B;MAEI,MAAM,6BAAsB,OAAtB,C;K;IAGV,iC;MAEI,MAAM,  
4CAAqC,uBAAqB,YAArB,8BAArC,C;K;ICIBV,8B;MC8CW,kBxGqBiD,oB;MwGM9C,Q;MAAA,OAAK,0B;M  
AAf,OAAU,cAAV,C;QAAU,mB;QACN,UAAU,sBAAM,CAAN,C;QACV,kBAakB,sBAAY,GAAZ,C;QAKFiD,U  
;QAJFnE,WxGyKJ,awGzKgB,GxGyKhB,EuG5OoB,CCmEkC,uBAAuB,CAAC,WAAY,mBAAY,GAAZ,CAiFhD,  
GDPJrC,CCoJqC,GAA6B,UajFjC,WaifiC,6DDpJnD,IAAM,CAAN,IvG4OpB,C;;MuG5OA,OCqEO,W;K;IC3Ey  
B,oC;MAAC,oC;K;;;;IF9CFrC,kD;MAyDI,SAAY,MAAK,OAAL,EAAc,SAAd,EAAyB,OAAzB,C;K;iFCzDhB,iC  
;MAuBmC,0B;QAAA,aAAuD,S;MActF,SAAY,MAAK,UAAAL,C;K;;;;I8CoBhB,qB;MAK0B,Q;MADtB,UAAmB,  
E;MACnB,wBAAsB,KAAtB,gB;QAAsB,aAAA,KAAtB,M;QAAK,IAAC,0BAAD,EAAO,2B;QACR,IAAI,IAAJ,I  
AAY,K;;MAEhB,OAAO,G;K;IAGX,+B;MAMgB,Q;MADZ,WAA0B,MAAa,MAAK,KAAL,C;MACvC,wBAAY,I  
AAZ,gB;QAAY,UAAA,IAAZ,M;QACI,IAAU,KAAy,gBAAe,GAAf,CAAtB,C;UACI,UAAK,GAAL,IAAY,MAA  
M,GAAN,C;;MAGpB,OAAO,S;K;qEC5DX,yB;MAAA,iB;MAAA,oB;QAOkD,OAAA,MAAW,KAAI,CAAJ,C;O  
;KAP7D,C;qEASA,yB;MAAA,iB;MAAA,oB;QAOkD,OAAA,MAAW,KAAI,CAAJ,C;O;KAP7D,C;qEASA,yB;M  
AAA,iB;MAAA,oB;QAOkD,OAAA,MAAW,KAAI,CAAJ,C;O;KAP7D,C;uEASA,yB;MAAA,iB;MAAA,oB;QAS  
mD,OAAA,MAAW,MAAK,CAAL,C;O;KAT9D,C;uEAWA,yB;MAAA,iB;MAAA,oB;QASmD,OAAA,MAAW,  
MAAK,CAAL,C;O;KAT9D,C;uEAWA,yB;MAAA,iB;MAAA,oB;QASmD,OAAA,MAAW,MAAK,CAAL,C;O;K  
AT9D,C;yEAWA,yB;MAAA,iB;MAAA,uB;QAKb+D,OAAA,MAAW,OAAM,CAAN,EAAS,CAAT,C;O;KAIB1E  
C;uEAoBA,yB;MAAA,0B;MAAA,oB;QAUmD,kBAAW,CAAX,C;O;KAVnD,C;uEAYA,yB;MAAA,0B;MAAA,  
oB;QASmD,kBAAW,CAAX,C;O;KATnD,C;uEAWA,yB;MAAA,0B;MAAA,oB;QAUmD,kBAAW,CAAX,C;O;K  
AVnD,C;yEAYA,yB;MAAA,4B;MAAA,oB;QAYoD,mBAAY,CAAZ,C;O;KAZpD,C;yEAcA,yB;MAAA,4B;MA  
AA,oB;QAYoD,mBAAY,CAAZ,C;O;KAZpD,C;yEAcA,yB;MAAA,4B;MAAA,oB;QAaoD,mBAAY,CAAZ,C;O;

KAbpD,C;yEAEa,yB;MAAA,4B;MAAA,uB;QAS+D,mBAAY,CAAZ,EAAe,CAAf,C;O;KAT/D,C;uEAWA,yB;M  
AAA,iB;MAAA,oB;QAQmD,OAAA,MAAW,MAAK,CAAL,C;O;KAR9D,C;qEAUa,yB;MAAA,iB;MAAA,oB;Q  
AUkD,OAAA,MAAW,KAAI,CAAJ,C;O;KAV7D,C;yEAYA,yB;MAAA,4B;MAAA,oB;QAcO,mBAAY,CAAZ,  
C;O;KAdpD,C;IAGBA,sB;MAcI,IAAI,QAAQ,GAAR,IAAe,SAAQ,GAA3B,C;QAAgC,OAAO,wCAAO,I;MAC9C  
,OAAO,IAAW,KAAI,CAAJ,CAAX,GAAoB,IAAW,KAAI,IAAJ,C;K;mEAG1C,yB;MAAA,iB;MAAA,oB;QAWi  
D,OAAA,MAAW,KAAI,CAAJ,C;O;KAX5D,C;yEAAa,yB;MAAA,4B;MAAA,oB;QAOoD,mBAAY,CAAZ,C;O;  
KAPpD,C;uEASA,yB;MAAA,0B;MAAA,oB;QAOmD,kBAAW,CAAX,C;O;KAPnD,C;uEASA,yB;MAAA,4B;M  
AAA,oB;QAgBmD,mBAAY,CAAZ,C;O;KAhBnD,C;uEAkBA,yB;MAAA,iB;MAAA,oB;QAUmD,OAAA,MAA  
W,MAAK,CAAL,C;O;KAV9D,C;yEAYA,yB;MAAA,iB;MAAA,oB;QAUoD,OAAA,MAAW,OAAM,CAAN,C;O  
;KAV/D,C;+EAYA,yB;MAAA,4B;MAAA,oB;QAUuD,mBAAY,CAAZ,C;O;KAVvD,C;IAYA,kB;MAQI,IAAI,IA  
AI,GAJ,KAAW,GAAf,C;QACI,OAAO,IAAW,OAAM,CAAN,C;;MAEtB,YAzBgD,MAAW,OAYBzC,CAzByC,  
C;MA0B3D,OAAW,QAAQ,CAAR,KAAa,GAAxB,GAA6B,KAA7B,GATC+C,MAAW,MASCb,CATCa,C;K;qEAy  
C9D,yB;MAAA,iB;MAAA,oB;QAUkD,OAAA,MAAW,KAAI,CAAJ,C;O;KAV7D,C;uEAYA,yB;MAAA,0B;MA  
AA,oB;QAWmD,kBAAW,CAAX,C;O;KAXnD,C;wEAca,yB;MAAA,iB;MAAA,uB;QAO6D,OAAA,MAAW,KAA  
AI,CAAJ,EAAO,CAAP,C;O;KAPxE,C;wEASA,yB;MAAA,iB;MAAA,uB;QAO6D,OAAA,MAAW,KAAI,CAAJ,  
EAAO,CAAP,C;O;KAPxE,C;uEAUa,yB;MAAA,iB;MAAA,oB;QAmD,OAAA,MAAW,MAAK,CAAL,C;O;KA  
b9D,C;qEAkBA,yB;MAAA,iB;MAAA,+B;QAayD,OAAA,MAAW,KAAI,SAAJ,EAAU,CAAV,C;O;KAbpE,C;uE  
AeA,yB;MAAA,iB;MAAA,+B;QAOsD,OAAA,MAAW,KAAI,SAAJ,EAAU,CAAV,C;O;KAPjE,C;iGAmBsD,yB;  
MAAA,iB;MAAA,4B;QAAQ,OAAA,MAAW,KAAI,SAAJ,C;O;KAAAnB,C;+EAaT,yB;MAAA,0B;MAAA,4B;QA  
AQ,kBAAW,SAAX,C;O;KAAR,C;iFAE7C,yB;MAAA,6C;MAAA,kC;QAK8D,OAAK,SAAL,SAAK,EAAc,IAAd  
,C;O;KALnE,C;IAkqC,4B;MACjC,gBAAO,CAAP,C;QADyC,OACrB,QAAP,CAAC,SAAM,C;WACpB,IAAK,  
QAAL,SAAK,CAAL,IAAgB,cAAQ,wCAAO,kBAA/B,C;QAFyC,OAeW,S;WACpD,kBAAQ,wCAAO,UAF,C;Q  
AHyC,OAGb,YAAU,SAAL,SAAK,C;;QAHc,OAI5B,OAAL,SAAK,CAAL,GAAgB,S;K;IAG5B,2B;MAKI,IAAK  
,QAAL,SAAK,CAAL,IAAgB,cAAQ,wCAAO,kBAA/B,C;QADwC,OACY,S;WACpD,kBAAQ,GAAR,C;QAFwC,  
OAEzB,wCAAO,U;;QACP,WAAc,UAAL,SAAK,CAAL,yBAAuB,YAAO,CAAX,GAAc,CAAd,GAAqB,EAAxC,  
E;QAHgB,OnDjC6B,MAAa,gBAaE,IAAf,C;;K;ImDuctF,6B;MAKI,IAAK,QAAL,SAAK,CAAL,IAAgB,cAAQ,w  
CAAO,kBAA/B,C;QAD0C,OACU,S;WACpD,kBAAQ,GAAR,C;QAF0C,OAE3B,CAAC,wCAAO,U;;QACR,WA  
Ac,UAAL,SAAK,CAAL,yBAAuB,YAAO,CAAX,GAAc,EAAc,GAAsB,CAAzC,E;QAHkB,OnD3c2B,MAAa,gB  
AAe,IAAf,C;;K;ImDkdtF,oC;MAUI,IAAK,QAAL,SAAK,CAAL,IAAmB,QAAL,EAAG,CAAnB,C;QADuD,OAC  
zB,wCAAO,I;WACrC,WAAM,SAAN,C;QAFuD,OAEzC,E;WACd,SAAK,SAAL,C;QAHuD,OAGrC,OAAL,SA  
AK,C;;QAHqC,OAI1B,SAAL,SAAK,C;K;IAIjC,+B;MAYI,uB;QAAW,MAAM,gCAAYB,yBAAzB,C;WACjB,gBA  
AO,UAAp,C;QAFyC,OAeJB,U;WACxB,gBAAO,WAAP,C;QAHyC,OAGjB,W;;QAHiB,OAIV,YAAvB,IAAW,O  
AAM,SAAN,CAAY,C;K;IAGnC,gC;MAYI,uB;QAAW,MAAM,gCAAYB,yBAAzB,C;WACjB,oD;QAF2C,+B;W  
AG3C,oD;QAH2C,+B;;QAAA,OAIz,uBAAvB,IAAW,OAAM,SAAN,CAAY,C;K;uEASnC,yB;MAAA,iB;MAAA  
,oB;QAOgD,OAAA,MAA6B,KAAZ,CAAY,C;O;KAP7E,C;uEASA,yB;MAAA,iB;MAAA,oB;QAOgD,OAAA,MA  
AA6B,KAAZ,CAAY,C;O;KAP7E,C;uEASA,yB;MAAA,iB;MAAA,oB;QAOgD,OAAA,MAA6B,KAAZ,CAAY,C  
;O;KAP7E,C;yEASA,yB;MAAA,iB;MAAA,oB;QASiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAT/E,C;yEAWA,y  
B;MAAA,iB;MAAA,oB;QASiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAT/E,C;yEAWA,yB;MAAA,iB;MAAA,o  
B;QASiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAT/E,C;2EAWA,yB;MAAA,iB;MAAA,uB;QAKB4D,OAAA,MA  
A6C,OAA1B,CAA0B,EAAZ,CAAY,C;O;KAIBzG,C;yEAoBA,yB;MAAA,0B;MAAA,oB;QAUiD,OAAyB,WAA  
Z,CAAY,C;O;KAV1E,C;yEAYA,yB;MAAA,0B;MAAA,oB;QASiD,OAAyB,WAAZ,CAAY,C;O;KAT1E,C;yEA  
WA,yB;MAAA,0B;MAAA,oB;QAUiD,OAAyB,WAAZ,CAAY,C;O;KAV1E,C;2EAYA,yB;MAAA,4B;MAAA,o  
B;QAYkD,OAA0B,YAAZ,CAAY,C;O;KAZ5E,C;2EAca,yB;MAAA,4B;MAAA,oB;QAYkD,OAA0B,YAAZ,CA  
AY,C;O;KAZ5E,C;2EAca,yB;MAAA,4B;MAAA,oB;QAakD,OAA0B,YAAZ,CAAY,C;O;KAb5E,C;2EAca,yB;  
MAAA,4B;MAAA,uB;QAS4D,OAAwC,YAA1B,CAA0B,EAAZ,CAAY,C;O;KATpG,C;yEAWA,yB;MAAA,iB;  
MAAA,oB;QAQiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAR/E,C;uEAUa,yB;MAAA,iB;MAAA,oB;QAUgD,O  
AAA,MAA6B,KAAZ,CAAY,C;O;KAV7E,C;2EAYA,yB;MAAA,4B;MAAA,oB;QAcK,D,OAA0B,YAAZ,CAAY,  
C;O;KAd5E,C;uEAga,yB;MAAA,mC;MAAA,0B;QAc6D,OAAmC,IAA7B,CAA6B,EAAZ,IAAY,C;O;KAdhG,

C;qEAgBA,yB;MAAA,iB;MAAA,oB;QAW+C,OAAA,MAA6B,KAAZ,CAAY,C;O;KAX5E,C;2EAaA,yB;MAAA,4B;MAAA,oB;QAOiD,OAAyB,WAAZ,CAAY,C;O;KAP1E,C;yEASA,yB;MAAA,0B;MAAA,oB;QAOiD,OAAyB,WAAZ,CAAY,C;O;KAP1E,C;yEASA,yB;MAAA,4B;MAAA,oB;QAgBiD,OAA0B,YAAZ,CAAY,C;O;KAhB3E,C;yEAKBA,yB;MAAA,iB;MAAA,oB;QAUiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAV/E,C;2EAYA,yB;MAAA,iB;MAAA,oB;QAUkD,OAAA,MAA+B,OAAZ,CAAY,C;O;KAVjF,C;iFAYA,yB;MA5hBA,4B;MA4hBA,oB;QAUqD,OA5hBE,YA4hBS,CA5hBT,C;O;KakhBvD,C;2EAYA,yB;MAAA,uC;MAAA,oB;QAQkD,OAAoB,MAAZ,CAAY,C;O;KARtE,C;uEAWA,yB;MAAA,iB;MAAA,oB;QAUgD,OAAA,MAA6B,KAAZ,CAAY,C;O;KAV7E,C;yEAYA,yB;MAAA,0B;MAAA,oB;QAWiD,OAAyB,WAAZ,CAAY,C;O;KAX1E,C;wEAeA,yB;MAAA,iB;MAAA,uB;QAO0D,OAAA,MAAW,KAAI,CAAJ,EAAO,CAAP,C;O;KAPrE,C;wEASA,yB;MAAA,iB;MAAA,uB;QAO0D,OAAA,MAAW,KAAI,CAAJ,EAAO,CAAP,C;O;KAPrE,C;yEAUA,yB;MAAA,iB;MAAA,oB;QAaiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAb/E,C;sEAmBA,yB;MAAA,iB;MAAA,+B;QAasD,OAAA,MAA8C,KAA1B,SA A0B,EAAZ,CAAY,C;O;KAbpG,C;uEAeA,yB;MAAA,iB;MAAA,+B;QAoD,OAAA,MAA8C,KAA1B,SAA0B,EAAZ,CAAY,C;O;KAPIG,C;kGAmBoD,yB;MAAA,iB;MAAA,4B;QAAQ,OAAA,MAAgC,KAAZ,SAAY,C;O;KA AxC,C;gFAaT,yB;MAAA,0B;MAAA,4B;QAAQ,OAA4B,WAAZ,SAAY,C;O;KAAPc,C;gFAE3C,yB;MAAA,6C;MAAA,kC;QAO8D,OAA0C,SAArC,SAaQc,EAAZ,IAAY,C;O;KAPxG,C;iFASA,yB;MAAA,6C;MAAA,kC;QA K4D,OAA0C,SAArC,SAaQc,EAAZ,IAAY,C;O;KALtG,C;oFAQA,yB;MAAA,iD;MAAA,4B;QAYmD,OAAW,W AAX,SAAW,C;O;KAZ9D,C;sFAcA,yB;MAAA,mD;MAAA,4B;QAYqD,OAAW,YAAX,SAAW,C;O;KAZhE,C;I AoBA,kB;MAUqC,OAAI,IAAI,CAAR,GAAY,CAAC,CAAD,OAAM,CAAIB,GAA0B,C;K;wEAE/D,yB;MAAA,i B;MAAA,uB;QAKoD,OAAA,MAAW,KAAI,CAAJ,EAAO,CAAP,C;O;KAL/D,C;wEAOA,yB;MAAA,iB;MAAA, uB;QAKoD,OAAA,MAAW,KAAI,CAAJ,EAAO,CAAP,C;O;KAL/D,C;mGAiBgD,yB;MAAA,mC;MAAA,4B;QA AQ,WAAI,SAAJ,C;O;KAAR,C;IAShB,+B;MAC5B,gBAAO,CAAP,C;QADoC,OACxB,E;WACZ,gBAAO,CAAP, C;QAFoC,OAExB,C;QAFwB,OAG5B,C;K;IAKZ,kB;MASuC,OAAI,eAAI,CAAR,GAAY,CAAD,aAAX,GAAM B,C;K;wEAE1D,gB;MAKuD,OAAI,kBAAK,CAAL,MAAJ,GAAY,CAAZ,GAAMb,C;K;wEAE1E,gB;MAKuD,O AAI,kBAAK,CAAL,MAAJ,GAAY,CAAZ,GAAMb,C;K;mGAYxB,yB;MAAA,mC;MAAA,4B;QAAQ,WAAI,SA AJ,C;O;KAAR,C;IASjB,+B;MAC7B,2BAAO,CAAP,C;QADqC,OACzB,E;WACZ,2BAAO,CAAP,C;QAFqC,OA EzB,C;QAFyB,OAG7B,C;K;IC5mCZ,4B;MAI4C,qBAAQ,S;K;IAEpD,4B;MAI2C,qBAAQ,S;K;IAEnD,+B;MAGi D,qBAAQ,wCAAO,kBAAf,IAAoC,cAAQ,wCAAO,kB;K;IAEpG,iC;MAGgD,qBAAQ,uCAAM,kBAAd,IAAmC,c AAQ,uCAAM,kB;K;IAEjG,6B;MAG+C,QAAC,qBAAD,IAAiB,CAAC,kB;K;IAEjE,+B;MAG8C,QAAC,uBAAD, IAAiB,CAAC,kB;K;IAGhE,iC;MAOI,QAAQ,S;MACR,IAAI,CAAC,IAAM,UAAP,KAAsB,CAAE,KAAK,CAAP, GAAC,UAApC,K;MACJ,IAAI,CAAC,IAAM,SAAP,KAAsB,CAAE,KAAK,CAAP,GAAC,SAAPc,K;MACJ,IAAI, CAAC,IAAM,SAAP,KAAsB,CAAE,KAAK,CAAP,GAAC,SAAPc,K;MACJ,IAAI,CAAC,IAAM,QAAP,KAAsB, CAAE,KAAK,CAAP,GAAC,QAAPc,K;MACJ,IAAI,CAAC,IAAM,KAAP,KAAsB,CAAE,KAAK,EAA7B,K;MA CJ,OAAO,C;K;kGAGX,yB;MAAA,4B;MAAA,4B;QAM2D,mBAAY,SAAZ,C;O;KAN3D,C;IAQA,0C;MAOI,YA TuD,YASvB,EAAf,aAAQ,CAAC,SAAD,IAAR,CAAE,CATuB,CASvD,I;K;IAEJ,sC;MAOI,OAAI,cAAQ,CAAZ, GAAe,CAAF,GAAsB,CAAE,IAAI,EAAJ,GAlB+B,sB;K;IAoB3D,qC;MAQI,oBAAS,CAAC,SAAD,IAAT,C;K;IA EJ,yC;MAaI,oBAAI,QAAJ,GAaiB,cAAK,EAAL,GAAqB,Q;K;IAG1C,0C;MAaI,oBAAI,EAAJ,GAAoB,QAAPB, GAAiC,cAAK,Q;K;IAG1C,mC;MAMI,OAAK,apDhEmD,uBoDgEnD,CAAL,GAA0B,apDjE6B,sBoDiE7B,CAA1 B,I;K;IAEJ,2C;MAMU,WAAW,SpDxEuC,c;MoDyEpD,e;QADJ,OACS,KA7E8C,YpDGA,sBoDHA,CA6E9C,I;Q ADT,OA5EuD,YA8E3C,IA9E2C,C;K;IAiF3D,4C;MAMU,UAAU,SpDpFuC,a;MoDqFnD,c;QADJ,OACS,KAAq B,sBpDpF0B,uBoDoF1B,CAArB,I;QADT,OAEGb,sBAAJ,GAAL,C;K;IAGpB,wC;MAOU,WAAW,SpD/FuC,c;M oDgGpD,e;QAAK,UAAAS,kBpDjGqC,sBoDiGrC,C;QADIB,OpDjG4C,MAAa,KAAK,UAAAS,GAAT,EoDkGvB,Cp DIguB,C;QoDmGID,aAAa,kBAAL,IAAK,C;QAFzB,OpDjG4C,MAAa,KAAK,UoDmG7C,CpDnG6C,EAAc,MA Ad,C;K;IoDsGIE,uC;MAOU,UAAU,SpD5GuC,a;MoD6GnD,c;QAAK,WAAa,iBpD5GkC,uBoD4GIC,C;QADtB, OpD7G4C,MAAa,KAAK,UoD8GhD,CpD9GgD,EAAc,IAAd,C;QoD+GID,YAAS,iBAAJ,GAAL,C;QAFrB,OpD7 G4C,MAAa,KAAK,UAAAS,KAAT,EoD+GrB,CpD/GqB,C;K;IoDkHIE,2C;MAaI,IAAI,CAAC,WAAa,EAAd,MA AqB,CAAzB,C;QACI,UAAU,SpD/HyC,a;QoDgInD,WAAW,SpD/HyC,c;QoDgIpD,aAAa,GAAL,IAAI,QAAR,GA AqB,IAAK,MAAK,CAAC,QAAD,IAAL,C;QACvC,cAAc,IAAK,IAAI,QAAT,GAAsB,GAAL,MAAK,CAAC,QA AD,IAAL,C;QACxC,OAAW,CAAC,WAAa,EAAd,MAAqB,CAAhC,GpDpIwC,MAAa,KAAK,UoDollB,MpDplk

B,EoDoIV,OpDpIU,CoDoI1D,GpDpIwC,MAAa,KAaK,UoDoIS,OpDpIT,EoDoIkB,MpDpIIB,C;;QoDsInD,Q;QA  
 AA,IAAI,CAAC,WAAa,EAAd,MAAqB,CAAzB,C;UAAA,OAA4B,S;;uBpDpLiB,uB;UoDoIP,apDrIM,sB;UoDqI5  
 C,OpDtIIC,MAAa,KAaK,kBAAc,MAAd,C;;QoDsI1D,W;;K;kFAKR,yB;MAAA,4C;MAAA,sC;QAaIE,6BAAW,  
 CAAC,QAAD,IAAX,C;O;KAbjE,C;qECvKA,kC;MAII,OAAO,SAA8B,MAAK,WAAL,C;K;uEAGzC,8C;MAII,O  
 AAO,SAA8B,MAAK,WAAL,EAakB,UAAIB,C;K;ICtZc,iC;MACI,gBAAH,IAAI,OAAO,EAAG,GAAE,IAAI,I  
 AAI,CAAC,CAAD,EAAl,EAaJ,CAAd,GAAyB,CAAhC,C;K;;IAKJ,sC;MACI,cAAO,QAAP,GAAkB,QAAQ,Q;  
 K;ICP9B,yC;K;;IAWA,+B;K;;4GAYa,yB;MAAA,gC;MAAA,yD;MAAA,sC;QAQI,OAAK,qBAAL,SAak,iB;O;  
 KART,C;ICPI,2B;MAAS,Q;MAAD,OAAwB,CAAvB,iEAAuB,Q;K;IAMhC,+B;MAAQ,iBAAU,SAaV,C;K;;;;I  
 CtB+B,4B;MACvC,8B;K;gEAAA,Y;MAAA,4B;K;2FAII,Y;MrGO4B,MAAM,yB;K;kCqGLtC,iB;MACI,OAAO,o  
 CAA0B,oBAAU,KAAM,OAAhB,C;K;oCAGRc,Y;MAC+B,gB;MAAA,8FAA0B,C;K;oCAEzD,Y;MAEI,OAAO,o  
 BAAQ,eAAR,C;K;;IAIyB,kC;MAAuB,sBAAc,MAAd,C;MACL,Q;MAAtD,4BAAmC,CAAmB,OAAZ,MAAY,W  
 AAAnB,kC;K;8FAAnC,Y;MAAA,gC;K;oDAEA,iB;MACW,cAAGB,W;MAAvB,OnEoEuD,MAAa,QmEpEpD,KnE  
 oEoD,EAAY,OAAZ,C;K;;ImEhEjC,0E;MAIvC,sBAAc,MAAd,C;MAFA,wC;MACA,8C;K;2CAEA,iB;MACI,IAA  
 I,0CAAJ,C;QAAsC,OAAO,K;MAC7C,OAAa,uCAAO,KAAP,CAAN,IAAuB,+BAAmB,KAAM,kBAAzB,C;K;iG  
 AGD,Y;MAAQ,6B;K;uDAEzC,iB;MACI,OAAO,0BAAmB,KAAnB,C;K;;IAIf,6B;MAAA,iC;MAAoC,sBAaOB,  
 MAApB,C;MACHC,4BAakC,S;K;+FAAlC,Y;MAAA,gC;K;qDAEA,iB;MAAgD,Y;K;2FAG5C,Y;MAAQ,MAAM  
 ,qCAA8B,6CAA9B,C;K;yCAEIB,iB;MAA4C,iBAAU,I;K;2CAEtD,Y;MAA+B,Q;K;;IAVnC,yC;MAAA,wC;QAA  
 A,uB;;MAAA,iC;K;IAaA,uB;K;yFACqC,Y;MxG0EY,MAAM,6BwG1EJ,oCxBG0EkC,WAA9B,C;K;4FwGzEf,Y;M  
 xGyES,MAAM,6BwGzED,uCxBGyE+B,WAA9B,C;K;+CwGvEnD,iB;MxGuE6C,MAAM,6BwGvEG,uCxBGUE2B,  
 WAA9B,C;K;mCwGrEnD,iB;MAA4C,iBAAU,I;K;qCAEtD,Y;MAA+B,Q;K;;oHCnE/B,qB;MAAQ,2B;K;;;;;  
 ;;;;;;;ICKZ,gE;MAMI,qBAAU,UAaV,EAAgC,OAAV,WAAU,CAAhC,EAA0C,gBAA1C,C;K;IAEJ,8B;M  
 AC2C,iC;K;IAE3C,kC;MAC+C,qBAAU,cAAA,KAAM,WAAN,CAAU,EAA8B,KAAM,UAApC,EAA+C,IAA/C,  
 C;K;IAE/C,2D;MAM0B,IAAN,I;MAAA,QAAM,QAAN,C;aACZ,I;;UAAA,K;aACA,K;;UAAK,K;;UAFY,K;;MA  
 AhB,oB;MAMA,OAAO,uBAAmB,IAAnB,EAaQc,OAAZ,WAAY,CAArC,EAA+C,SAA/C,EAA0D,KAA1D,C;K;  
 IAGX,kC;MAEI,OAAA,uCAAgB,K;K;IAEpB,8C;MAEI,OAAA,uCAAgB,mBAAU,IAAV,C;K;IAEpB,8C;MAEI,  
 OAAA,uCAAgB,mBAAU,IAAV,C;K;IAEpB,kD;MAEI,OAAA,uCAAgB,uBAAc,IAAd,C;K;IC/CI,8D;MACpB,s  
 C;MACA,sC;MACA,kD;K;mEFAA,Y;MAAA,gC;K;kEACA,Y;MAAA,+B;K;yEACA,Y;MAAA,sC;K;iCAEA,iB;  
 MACI,0CACQ,wBAAc,KAAM,WAAPB,CADR,IAC0C,uBAAa,KAAM,UAAAnB,CAD1C,IAC0E,0BAaOB,KAA  
 M,iB;K;mCAEXG,Y;MACI,SAAC,CAAW,SAAX,eAAW,CAAX,GAAwB,EAAXB,QAAuC,SAaV,cAAU,CAAU,  
 C,IAAD,IAAsD,EAAtD,QAA4E,SAAjB,qBAaiB,CAA5E,I;K;mCAEJ,Y;MACKB,UACO,M;MADrB,aAAc,2D;M  
 AEV,cAAU,IAAV,C;QAA6B,SAAX,eAAW,W;WAC7B,IAAA,MAAO,WAAP,S;QAAoC,SAAP,MAAO,W;;QA  
 C5B,+B;MAHZ,2B;MAMA,WACQ,cAAU,UAAa,GAAyB,EAazB,GACe,eAAV,cAAU,EAAa,IAAb,EAAmB,GA  
 AnB,EAAwB,GAAxB,C;MACnB,eAAmB,qBAAJ,GAAsB,GAAtB,GAA+B,E;MAE9C,OAAO,iBAAiB,IAAjB,G  
 AAwB,Q;K;;IAIvC,wB;MAAA,4B;MACI,4BAAwC,I;MACxC,2BAAgD,W;MACHD,kCAAyC,K;K;0FAFzC,Y;M  
 AAA,gC;K;yFACA,Y;MAAA,+B;K;gGACA,Y;MAAA,sC;K;sCACA,Y;MAAkC,gB;K;;IAJtC,oC;MAAA,mC;Q  
 AAA,kB;;MAAA,4B;K;IC7BsC,oE;MACIC,0B;MACA,wC;MACA,kC;MACA,oC;K;seAHA,Y;MAAA,0B;K;6E  
 ACA,Y;MAAA,iC;K;0EACA,Y;MAAA,8B;K;2EACA,Y;MAAA,+B;K;4CAEA,Y;MAAkC,gB;K;;8CANtC,Y;MA  
 CI,gB;K;8CADJ,Y;MAEI,uB;K;8CAFJ,Y;MAGI,oB;K;8CAHJ,Y;MAII,qB;K;gDAJJ,kD;MAAA,8BACI,kCADJ,E  
 AEI,uDAFJ,EAGI,8CAHJ,EAIL,iDAJ,C;K;4CAAA,Y;MAAA,c;MACI,qD;MACA,4D;MACA,yD;MACA,0D;MA  
 JJ,a;K;0CAAA,iB;MAAA,4IACI,oCADJ,IAEI,kDAFJ,IAGI,4CAHJ,IAII,8CAJJ,I;K;ICAA,4B;MAAA,gC;MAEI,g  
 BACe,wBAAoB,MAAPB,EAA6D,KAA7D,EAAoE,gCAApE,C;MAEf,mBACkB,wBAAoB,MAAPB,EAAgE,QA  
 AhE,EAA0E,mCAA1E,C;MAEIB,oBACmB,+B;MAEnB,oBACmB,wBAAoB,OAApB,EAAkE,SAaIE,EAA6E,oC  
 AA7E,C;MAEnB,iBACgB,wBAAoB,MAAPB,EAA8D,MAA9D,EAAsE,iCAAtE,C;MAEhB,kBACiB,wBAAoB,M  
 AAPB,EAA+D,OAA/D,EAAwE,kCAAxE,C;MAEjB,gBACe,wBAAoB,MAAPB,EAA6D,KAA7D,EAAoE,gCAAp  
 E,C;MAEf,kBACiB,wBAAoB,MAAPB,EAA+D,OAA/D,EAAwE,kCAAxE,C;MAEjB,mBACkB,wBAAoB,MAAP  
 B,EAAgE,QAaHE,EAA0E,mCAA1E,C;MAEIB,kBACiB,wBAAoB,KAAPB,EAAIE,OAAjE,EAA0E,kCAA1E,C;  
 MAEjB,mBACkB,wBAAoB,MAAPB,EAAgE,QAaHE,EAA0E,mCAA1E,C;MAEIB,sBACqB,wBAAoB,KAAPB,  
 EAAkE,WAAIE,EAA+E,sCAA/E,C;MAErB,yBACwB,wBAAoB,KAAPB,EAAqE,cAArE,EAAqF,yCAArF,C;MA

ExB,sBACqB,wBAAoB,WAApB,EAAwE,WAAxE,EAAqF,sCAArF,C;MAErB,sBACqB,wBAAoB,SAApB,EAA s  
E,WAAtE,EAAMf,sCAAnF,C;MAErB,uBACsB,wBAAoB,UAApB,EAAwE,YAAxE,EAA sF,uCAAtF,C;MAEtB,  
qBACoB,wBAAoB,UAApB,EAA sE,UAA tE,EA AkF,qCAAI F,C;MAEpB,sBACqB,wBAAoB,KAApB,EA AkE,W  
AAIE,EAA+E,sCAA/E,C;MAErB,uBACsB,wBAAoB,YAApB,EAA0E,YAA1E,EAAwF,uCAAx F,C;MAEtB,wBA  
CuB,wBAAoB,YAApB,EAA2E,AAA3E,EAA0F,wCAA1F,C;K;IAMkB,qE;MAAA,qB;QAAE,OzE/DD,OyE+DU,  
EAAT,KAAiB,UAAjB,IAAkC,EAAY,OAAf,KAA0B,a;O;K;+CAJpG,iB;MAE2B,Q;MAAhB,U;MAAA,KAAgB,  
OAAhB,eAAgB,CAAI,KAAJ,CAAhB,U;QAAA,a;;QACH,aAAa,wBAAoB,QAApB,EAA+D,kBAA/D,EACoB,m  
DADpB,C;QAEg,eAAhB,UAAqC,M;QAHIC,SAIH,M;;MAJJ,a;K;IA7D+E,8C;MAAE,6B;K;IAGO,iD;MAAE,0B  
;K;IAME,kD;MAAE,8B;K;IAGZ,+C;MAAE,6B;K;IAGC,gD;MAAE,6B;K;IAGR,8C;MAAE,6B;K;IAGI,gD;MA  
AE,6B;K;IAGC,iD;MAAE,6B;K;IAGH,gD;MAAE,yB;K;IAGD,iD;MAAE,6B;K;IAGM,oD;MAAE,mC;K;IAGO,u  
D;MAAE,gC;K;IAGL,oD;MAAE,6B;K;IAGJ,oD;MAAE,6B;K;IAGE,qD;MAAE,8B;K;IAGR,mD;MAAE,4B;K;I  
AGJ,oD;MAAE,6B;K;IAGQ,qD;MAAE,8B;K;IAGC,sD;MAAE,+B;K;;IA5DvH,wC;MAAA,uC;QAAA,sB;;MAA  
A,gC;K;;ICCA,2B;MAEW,Q;MAAA,IAAI,KAA Y,SAAQ,MAAR,CAAhB,C;QACH,kBAAW,MAAX,C;;QAEA,k  
BAAW,MAAX,C;;MAHJ,W;K;IAOJ,8B;MAC4E,QAAM,QAAS,OAAf,C;aACxE,C;UADwE,OACnE,WAAW,SA  
AS,CAAT,CAAX,C;aACL,C;UAFwE,OAEnE,+B;;UAFmE,OAGhE,iB;;K;IAGZ,oC;MAEU,IAAN,I;MAAA,Q1E  
hB0C,O0EgB3B,CAAf,C;aACI,Q;UAA6B,OAAjB,8BAAiB,Y;UAA7B,K;aACA,Q;UAA Y,OAAI,CAAY,CjEbhC,  
GiEamC,CAAf,MAAkC,CAAtC,GAAyC,8BAAiB,SAA1D,GAAwE,8BAAiB,Y;UAArG,K;aACA,S;UAA8B,OAA  
jB,8BAAiB,a;UAA9B,K;aACA,U;UAA+B,OAAjB,8BAAiB,eAAgB,CAAY,OAA5B,C;UAA/B,K;;UAGQ,6B;YA  
AsC,OAAjB,8BAAiB,kB;eACtC,0B;YAAmC,OAAjB,8BAAiB,e;eACnC,0B;YAAmC,OAAjB,8BAAiB,e;eACnC,  
2B;YAAoC,OAAjB,8BAAiB,gB;eACpC,yB;YAAkC,OAAjB,8BAAiB,c;eAClC,0B;YAAmC,OAAjB,8BAAiB,e;e  
ACnC,2B;YAAoC,OAAjB,8BAAiB,gB;eACpC,4B;YAAqC,OAAjB,8BAAiB,iB;eACrC,6B;;eACA,sB;YAAkC,O  
AAjB,8BAAiB,W;;YAE9B,kBAAkB,MAAA,gBA Ae,CAAf,CAAkB,Y;YAE7C,oBAAgB,MAAhB,C;cAAiD,OAAj  
B,8BAAiB,S;iBACjD,oBAAgB,KA AhB,C;cAAGD,OAAjB,8BAAiB,e;;cAE5C,cAA0B,W;cAC1B,kBAAW,OAA  
X,C;;;UAXBxB,K;;MAAA,W;K;IAGCJ,4B;MAMW,Q;MAJP,IAAI,WAAW,MAAf,C;QAA6B,OAAO,8BAAiB,Y;  
;MAErD,eAA sB,MAAY,W;MAE3B,IAAI,gBAAJ,C;QACH,IAAI,QAAS,SAAT,QAAJ,C;UACI,aAAa,qBAAiB,M  
AAjB,C;UACb,oBAAsB,M;UActB,a;;UAES,OAAT,QAAS,S;;QAGb,4BAAiB,MAAjB,C;;MATJ,W;K;ICrCJ,0B;  
MAIL,sBAAY,C;K;qEACHB,4B;MAIkE,iBAAY,KAAZ,C;K;2EAEIE,qB;MAI8D,gB;K;ICIDb,2C;MAC7C,qBAA  
wC,Q;K;iDAExC,Y;MACmB,Q;MAAA,yB;MAAA,iB;QA Ae,MAAM,6BAAsB,0CAAtB,C;;MAApC,eAAe,I;MA  
Cf,qBAAc,I;MACd,OAAO,QAAS,W;K;;;ICLa,kD;MADrC,e;MACsC,0B;MAAyB,gB;MAD/D,iB;MAAA,uB;K;  
IAAA,mC;MAAA,sC;O;MAEI,qEAGW,CAHX,EAGc,IAHd,C;MAKA,iFAGiB,CAHjB,EAGoB,IAHpB,C;MAKA  
,iFAGiB,CAHjB,EAGoB,IAHpB,C;MAKA,iFAGiB,CAHjB,EAGoB,IAHpB,C;MAKA,+EAGgB,CAHhB,EAGmB  
,IAHnB,C;MAKA,yEAGa,CAHb,EAGgB,IAHhB,C;MAKA,iFAGiB,CAHjB,EAGoB,IAHpB,C;MAKA,6EAGe,C  
AHf,EAGkB,IAHIB,C;MAKA,6FAGuB,CAHvB,EAG0B,IAH1B,C;MAKA,yFAGqB,CAHrB,EAGwB,IAHxB,C;  
MAKA,4EAGc,EAHd,EAGkB,IAHIB,C;MAKA,0EAGa,EAHb,EAGiB,IAHjB,C;MAKA,gFAGgB,EAHhB,EAGo  
B,IAHpB,C;MAKA,8EAGe,EAHf,EAGmB,IAHnB,C;MAKA,wFAGoB,EAHpB,EAGwB,IAHxB,C;MAKA,gEAG  
Q,EAHR,EAGY,IAHZ,C;MAKA,8DAGO,EAHP,EAGW,IAHX,C;MAKA,wEAGY,EAHZ,EAGgB,IAHhB,C;MA  
KA,oEAGU,EAHV,EAGc,IAHd,C;MAKA,kFAGiB,EAHjB,EAGqB,IAHrB,C;MAKA,oFAGkB,EAHIB,EAGsB,I  
AHtB,C;MAKA,gFAGgB,EAHhB,EAGoB,IAHpB,C;MAKA,4FAGsB,EAHtB,EAG0B,IAH1B,C;MAKA,oFAGk  
B,EAHIB,EAGsB,IAHtB,C;MAKA,wEAGY,EAHZ,EAGgB,IAHhB,C;MAKA,gFAGgB,EAHhB,EAGoB,IAHpB,  
C;MAKA,gFAGgB,EAHhB,EAGoB,IAHpB,C;MAKA,0EAGa,EAHb,EAGiB,IAHjB,C;MAKA,oGAG0B,EAH1B,  
EAG8B,IAH9B,C;MAKA,gGAGwB,EAHxB,EAG4B,IAH5B,C;MAUA,oC;K;;IA3JA,+C;MAAA,yB;MAAA,uC;  
K;;IAKA,qD;MAAA,yB;MAAA,6C;K;;IAKA,qD;MAAA,yB;MAAA,6C;K;;IAKA,qD;MAAA,yB;MAAA,6C;K;;I  
AKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,iD;MAAA,yB;MAAA,yC;K;;IAKA,qD;MAAA,yB;MAAA,6C;K;;IAK  
A,mD;MAAA,yB;MAAA,2C;K;;IAKA,2D;MAAA,yB;MAAA,mD;K;;IAKA,yD;MAAA,yB;MAAA,iD;K;;IAKA,  
kD;MAAA,yB;MAAA,0C;K;;IAKA,iD;MAAA,yB;MAAA,yC;K;;IAKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,mD;  
MAAA,yB;MAAA,2C;K;;IAKA,wD;MAAA,yB;MAAA,gD;K;;IAKA,4C;MAAA,yB;MAAA,oC;K;;IAKA,2C;M  
AAA,yB;MAAA,mC;K;;IAKA,gD;MAAA,yB;MAAA,wC;K;;IAKA,8C;MAAA,yB;MAAA,sC;K;;IAKA,qD;MA  
AA,yB;MAAA,6C;K;;IAKA,sD;MAAA,yB;MAAA,8C;K;;IAKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,0D;MAAA,

yB;MAAA,kD;K;;IAKA,sD;MAAA,yB;MAAA,8C;K;;IAKA,gD;MAAA,yB;MAAA,wC;K;;IAKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,iD;MAAA,yB;MAAA,yC;K;;IAKA,8D;MAAA,yB;MAAA,sD;K;;IAKA,4D;MAAA,yB;MAAA,oD;K;8CAKA,gB;MAG2D,OAAK,iBAAL,IAAK,CAAL,KAA2B,IAAK,c;K;IAE3F,kC;MAAA,sC;K;uDACL,oB;MAEQ,IADE,QACF,IAAG,CAAH,IADE,QACF,IAAM,EAAN,C;QADJ,OACgB,sBAAS,QAAT,C;WACZ,IAFE,QAEF,IAAG,EAAH,IAFE,QAEF,IAAO,EAAP,C;QAFJ,OAEiB,sBAAS,WAAW,CAAX,IAAT,C;;QACL,MAAM,gCAAYB,eAAY,QAAZ,qBAAZB,C;K;;IAL1B,8C;MAAA,yB;MAAA,6C;QAAA,4B;;MAAA,sC;K;;IA7JJ,+B;MAAA,+yC;K;;IAAA,oC;MAAA,a;AAAA,Y;UAAA,4C;aAAA,kB;UAAA,kD;aAAA,kB;UAAA,kD;aAAA,kB;UAAA,kD;aAAA,iB;UAAA,iD;aAAA,c;UAAA,8C;aAAA,kB;UAAA,kD;aAAA,gB;UAAA,gD;aAAA,wB;UAAA,wD;aAAA,sB;UAAA,sD;aAAA,e;UAAA,+C;aAAA,c;UAAA,8C;aAAA,iB;UAAA,iD;aAAA,gB;UAAA,gD;aAAA,qB;UAAA,qD;aAAA,S;UAAA,yC;aAAA,Q;UAAA,wC;aAAA,a;UAAA,6C;aAAA,W;UAAA,2C;aAAA,kB;UAAA,kD;aAAA,mB;UAAA,mD;aAAA,iB;UAAA,iD;aAAA,uB;UAAA,uD;aAAA,mB;UAAA,mD;aAAA,a;UAAA,6C;aAAA,iB;UAAA,iD;aAAA,iB;UAAA,iD;aAAA,c;UAAA,8C;aAAA,2B;UAAA,2D;aAAA,yB;UAAA,yD;;UAAA,6D;;K;;ICKiD,2C;uBAA+B,O;;K;;IAC5E,8C;MAAA,kE;MAAuB,qCAAK,IAAL,C;MAAvB,Y;K;ICD8B,gC;MAe9B,gBAAiC,YAAY,SAAhB,GAA2B,OAA3B,GAAwC,E;K;uFAGjE,Y;M AAQ,OAAO,aAAY,O;K;yCAE/B,iB;MACW,gBAAP,a;MjGqGG,Q;MAAA,IiGrGc,KjGqGV,IAAS,CAAT,IiGrGU,KjGqGI,IAAS,2BAA3B,C;QAAA,OAAc,qBiGrGxB,KjGqGwB,C;;QiGrGf,MAAM,8BAA0B,mCAAYB,WAAzB,MAA1B,C;;MAAhC,W;K;kDAEJ,gC;MAAgF,OAAA,atG0NY,WsG1NK,UtG0NL,EsG1NiB,QtG0NjB,C;K;6CsGxN5F,iB;MACI,qCAAU,KAAV,C;MACA,OAAO,I;K;6CAGX,iB;MACI,iBAAGB,SAAN,KAAM,C;MACHB,OAAO,I;K;6CAGX,uC;MACI,OAAA,IAAK,qBAAY,wBAAS,MAArB,EAA6B,UAA7B,EAAyC,QAAzC,C;K;sCAET,Y;MAayB,UAEK,M;MAL1B,eAAe,E;MACf,YAAY,aAAO,OAAP,GAAgB,CAAhB,I;MACZ,OAAO,SAAS,CAAhB,C;QACI,UAAU,0BAAO,YAAP,EAAO,oBAAP,Q;QACV,IAAQ,eAAJ,GAAl,CAAJ,IAAwB,SAAS,CAArC,C;UACI,WAAW,0BAAO,cAAP,EAAO,sBAAP,U;UACX,IAAS,gBAAL,IAAK,CAAT,C;YACI,WAAW,+BAAW,iBAAX,wBAAkB,gBAAIB,C;;YAEX,WAAW,+BAAW,gBAAX,wBAAiB,iBAAjB,C;;UAGf,gCAAY,GAAZ,C;;MAGR,gBAAS,Q;MACT,OAAO,I;K;6CAGX,iB;MAOI,iBAAGB,SAAN,KAAM,C;MACHB,OAAO,I;K;6CAGX,iB;MAQI,iBAAU,K;MACV,OAAO,I;K;6CAGX,iB;MAQI,iBAAGB,eAAN,KAAM,C;MACHB,OAAO,I;K;6CAGX,iB;MAC2C,2BAAO,KAAP,C;K;6CAE3C,iB;MAOI,gBAAA,IAAK,SAAL,IAAe,wBAAS,MAAxB,C;MACA,OAAO,I;K;uCAGX,Y;MAU6B,kB;K;qDAE7B,2B;K;8CAcA,kB;MAO0C,OAAA,IAAY,SAAY,SAAQ,MAAR,C;K;8CAEIE,8B;MAQ2D,OAAA,IAAY,SAAY,SAAQ,MAAR,EAAgB,UAAhB,C;K;kDAEnF,kB;MAQ8C,OAAA,IAAY,SAAY,aAAY,MAAZ,C;K;kDAEtE,8B;MASI,IAAI,MhGuGwC,YAAU,CgGvGID,IAAoB,aAAa,CAArC,C;QAAwC,OAAO,E;MAC/C,OAAO,IAAY,SAAY,aAAY,MAAZ,EAAoB,UAApB,C;K;4CAGnC,wB;MAWI,oCAAa,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MAEb,gBAAS,atG4C+E,WsG5C9D,CtG4C8D,EsG5C3D,KtG4C2D,CsG5C/E,YAA6B,KAA7B,IAAqC,atGyC2B,WsGzCV,KtGyCU,C;MsGxCzE,OAAO,I;K;6CAGX,wB;MAQI,oCAAa,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MAEb,gBAAS,atG8B+E,WsG9B9D,CtG8B8D,EsG9B3D,KtG8B2D,CsG9B/E,uBAA6B,kBAA7B,IAAqC,atG2B2B,WsG3BV,KtG2BU,C;MsG1BzE,OAAO,I;K;6CAGX,wB;MAUI,oCAAa,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MAEb,gBAAS,atGc+E,WsGd9D,CtGc8D,EsGd3D,KtGc2D,CsGd/E,GAAMc,eAAN,KAAM,CAAnC,GAAsD,atGWU,WsGXO,KtGWP,C;MsGvzE,OAAO,I;K;6CAGX,wB;MAA1,oCAAa,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MAEb,gBAAS,atGL+E,WsGK9D,CtGL8D,EsGK3D,KtGL2D,CsGK/E,GAAMc,SAAN,KAAM,CAAnC,GAAGD,atGRgB,WsGQC,KtGRD,C;MsGSzE,OAAO,I;K;6CAGX,wB;MAWI,oCAAa,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MAEb,gBAAS,atGtB+E,WsGsB9D,CtGtB8D,EsGsB3D,KtGtB2D,CsGsB/E,GAAMc,SAAN,KAAM,CAAnC,GAAGD,atGzBgB,WsGyBC,KtGzBD,C;MsG0BzE,OAAO,I;K;6CAGX,wB;MACuD,2BAAO,KAAP,EAAC,KAAc,C;K;6CAEvD,wB;MAUI,oCAAa,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MAEb,eAAe,wBAAS,M;MACxB,gBAAc,IAAK,StG1CqE,WsG0CpD,CtG1CoD,EsG0CjD,KtG1CiD,CsG0C1E,GAAC,K,QAAIC,GAA6C,IAAK,StG7CS,WsG6CQ,KtG7CR,C;MsG8CzE,OAAO,I;K;gDAGX,qB;MAcI,IAAI,YAAY,CAAhB,C;QACI,MAAM,gCAAYB,0BAAuB,SAAvB,MAAZB,C;;MAGV,IAAI,aAAa,WAAjB,C;QACI,gBAAS,atGjE2E,WsGiE1D,CtGjE0D,EsGiEvD,StGjEuD,C;;QsGmEpF,aAAU,WAAV,MAAuB,SAAvB,M;UACI,qCAAU,CAAV,C;;K;gDAKZ,sB;MAQI,oCAAa,4BAAmB,UAAAnB,EAA+B,WAA/B,C;MAEb,OAAO,atGtFkE,WsGsFjD,UtGtFiD,C;K;gDsGyF7E,gC;MAQI,oCAAa,4BAAmB,UAAAnB,EAA+B,QAA/B,EAAyC,WAAzC,C;MAEb,OAAO,atGhGiF,WsGgGhE,UtGhGgE,EsGgGpD,QtGhGoD,C;K;yCsGmG5F,Y;K;uCACa,Y;MAAkC,oB;K;o



CAEIC,Y;MAOI,gBAAS,E;MACT,OAAO,I;K;0CAGX,wB;MAQI,oCAAA,2BAAkB,KAAIB,EAAyB,WAAzB,C; MAEb,gBAAS,atGxI+E,WsGwI9D,CtGxI8D,EsGwI3D,KtGxI2D,CsGwI/E,uBAA6B,kBAA7B,IAAqC,atG3I2B,W sG2IV,QAAQ,CAAR,ItG3IU,C;K;+CsG8I7E,uC;MAYI,yBAAkB,UAAIB,EAA8B,QAA9B,EAAwC,WAAxC,C; MAEA,gBAAC,IAAK,StGzJqE,WsGyJpD,CtGzJoD,EsGyJjD,UtGzJiD,CsGyJ1E,GAAuC,KAAvC,GAA+C,IAAK, StG5JO,WsG4JU,QtG5JV,C;MsG6JzE,OAAO,I;K;kDAGX,wC;MACI,IAAI,aAAa,CAAb,IAAkB,aAAa,MAAnC, C;QACI,MAAM,8BAA0B,iBAAc,UAAAd,kBAAMC,MAA7D,C;MAEV,IAAI,aAAa,QAAjB,C;QACI,MAAM,gC AAYB,gBAAa,UAAb,qBAAqC,QAArC,MAAzB,C;K;+CAId,iB;MAYI,oCAAA,2BAAkB,KAAIB,EAAyB,WAAz B,C;MAEb,gBAAS,atGpL+E,WsGoL9D,CtGpL8D,EsGoL3D,KtGpL2D,CsGoL/E,GAA6B,atGvLmC,WsGuLlB,Q AAQ,CAAR,ItGvLkB,C;MsGwLzE,OAAO,I;K;kDAGX,gC;MAWI,yBAAkB,UAAIB,EAA8B,QAA9B,EAAwC, WAAxC,C;MAEA,gBAAS,atGrM+E,WsGqM9D,CtGrM8D,EsGqM3D,UtGrM2D,CsGqM/E,GAAkC,atGxM8B, WsGwMb,QtGxMa,C;MsGyMzE,OAAO,I;K;kDAGX,gE;MAc+C,iC;QAAA,oBAAyB,C;MAAG,0B;QAAA,aAAk B,C;MAAG,wB;QAAA,WAAgB,IAAK,O;MAKIF,IACf,I;MALhB,oCAAA,4BAAMB,UAAAnB,EAA+B,QAA/B,E AAYC,WAAzC,C;MACb,oCAAA,4BAAMB,iBAAnB,EAAcS,oBAAoB,QAApB,GAA+B,UAA/B,IAAtC,EAAiF, WAAy,OAA7F,C;MAEb,eAAe,iB;MACf,iBAAc,UAAAd,UAA+B,QAA/B,U;QACI,YAAY,eAAZ,EAAy,uBAAZ, UAA0B,yBAAO,KAAP,C;K;kDAIIC,uC;MAcI,iBAAGB,iBAAN,KAAM,EAAe,UAAf,EAA2B,QAA3B,C;MACH B,OAAO,I;K;kDAGX,uC;MAYI,gBAAgB,KAAM,W;MAcTb,oCAAA,4BAAMB,UAAAnB,EAA+B,QAA/B,EAAy C,SAAU,OAAAnD,C;MAEb,iBAAU,StGIQ8E,WsGkQ1D,UtGIQ0D,EsGkQ9C,QtGIQ8C,C;MsGmQxF,OAAO,I;K; kDAGX,8C;MAgBI,oCAAA,4BAAMB,KAAAnB,EAA0B,IAAK,OAA/B,C;MAEb,gBAAS,atGxR+E,WsGwR9D,Ct GxR8D,EsGwR3D,KtGxR2D,CsGwR/E,GAAmC,iBAAN,KAAM,EAAe,UAAf,EAA2B,QAA3B,CAAnC,GAA0E, atG3RV,WsG2R2B,KtG3R3B,C;MsG4RzE,OAAO,I;K;kDAGX,8C;MAgBI,oCAAA,4BAAMB,KAAAnB,EAA0B,W AA1B,C;MAEb,gBAAgB,KAAM,W;MAcTb,oCAAA,4BAAMB,UAAAnB,EAA+B,QAA/B,EAAyC,SAAU,OAAAnD ,C;MAEb,gBAAS,atGjT+E,WsGiT9D,CtGjT8D,EsGiT3D,KtGjT2D,CsGiT/E,GAA6B,StGjTkD,WsGiT9B,UtGjT8 B,EsGiTIB,QtGjTkB,CsGiT/E,GAAyE,atGpTT,WsGoT0B,KtGpT1B,C;MsGqTzE,OAAO,I;K;;IALiBX,6C;MAAA, uD;MAKoC,2B;MALpC,Y;K;IAQA,8C;MAAA,uD;MAC4C,0BAAK,OAAQ,WAAb,C;MAD5C,Y;K;IAGA,qC;M AAA,uD;MACuB,0BAAK,EAAL,C;MADvB,Y;K;2EA4hBJ,qB;MAOgE,OAAA,SAAK,Q;K;uEAERe,mC;MAQ+ E,SAAK,aAAI,KAAJ,EAAW,KAAx,C;K;+EAEPf,kD;MAaI,OAAA,SAAK,kBAAS,UAAAT,EAAqB,QAArB,EA A+B,KAA/B,C;K;+EAET,4B;MAY6E,OAAA,SAAK,kBAAS,KAAT,C;K;qFAEIF,2C;MAWoG,OAAA,SAAK,qB AAY,UAAZ,EAAwB,QAAxB,C;K;uFAEZg,2E;MAe2E,iC;QAAA,oBAAyB,C;MAAG,0B;QAAA,aAAkB,C;MA AG,wB;QAAA,WAAgB,SAAK,O;MAC7I,SAAK,qBAAy,WAAZ,EAAyB,iBAAzB,EAA4C,UAA5C,EAAwD,Q AAxD,C;K;qFAET,kD;MAeI,OAAA,SAAK,qBAAy,KAAZ,EAAmB,UAAAnB,EAA+B,QAA/B,C;K;uFAET,kD;M AaI,OAAA,SAAK,qBAAy,KAAZ,EAAmB,UAAAnB,EAA+B,QAA/B,C;K;qFAET,yD;MAiBI,OAAA,SAAK,qBA AY,KAAZ,EAAmB,KAAAnB,EAA0B,UAA1B,EAAcS,QAAtC,C;K;uFAET,yD;MAiBI,OAAA,SAAK,qBAAy,K AAZ,EAAmB,KAAAnB,EAA0B,UAA1B,EAAcS,QAAtC,C;K;qFvGhsBT,qB;MAMoD,OA6BW,8BAAY,cAfrB,Y AAY,CAAZ,C;K;yFAZtD,qB;MAYsD,OAeS,8BAAY,cAfrB,YAAY,CAAZ,C;K;iFAEtD,qB;MAaoD,OAAW,8B AAY,c;K;qFAE3E,yB;MAAA,uD;MAAA,4B;QAMoD,+B;O;KANpD,C;IAQA,kC;MAYI,gBAiB2D,8BAAY,c;M AhBvE,OAAW,SAAU,OAAV,GAAmB,CAAvB,GAA0B,SAA1B,GAAoC,qBAAU,CAAV,C;K;iFAG/C,qB;MAao D,OAAW,8BAAY,c;K;IAE3E,kC;MAU+C,mC;K;IAE/C,oC;MAGoD,QAAQ,cAAA,sCAAK,mBAAL,EAAyB,sC AAK,mBAA9B,CAAR,6B;K;IAEpD,mC;MAGmD,QAAQ,cAAA,sCAAK,kBAAL,EAAwB,sCAAK,kBAA7B,CA AR,6B;K;IAO/C,iC;MAAQ,OAAA,oCAAA,iBAAQ,2BAAR,C;K;IAEzB,8B;MAOI,IAAI,YAAO,GAAX,C;QACI, OAAO,I;;MAEX,OAAO,gCAA8C,mD;K;IAGzD,6B;MAUI,IAAI,CAAQ,kBAAK,GAAL,CAAR,iCAAoB,CAAQ, kBAAK,EAAL,CAAR,6BAAxB,C;QACI,OAAO,I;;MAEX,IAAI,YAAO,GAAX,C;QACI,OAAO,K;;MAEX,OAA O,uB;K;IAGX,oC;MAUI,IAAI,CAAQ,kBAAK,GAAL,CAAR,iCAAoB,CAAQ,kBAAK,EAAL,CAAR,6BAApB,I AAwC,CAAQ,kBAAK,EAAL,CAAR,6BAA5C,C;QACI,OAAO,I;;MAEX,IAAI,YAAO,GAAX,C;QACI,OAAO,K ;MAGX,OAAO,0BAAiB,uB;K;IAG5B,4B;MASI,IAAI,CAAQ,kBAAK,EAAL,CAAR,6BAAJ,C;QACI,OAAO,I;; MAEX,IAAI,YAAO,GAAX,C;QACI,OAAO,K;;MAEX,OAAO,sB;K;IAGX,gC;MAUI,IAAI,CAAQ,kBAAK,EAAL,CAAR,6BAAJ,C;QACI,OAAO,I;;MAEX,IAAI,YAAO,GAAX,C;QACI,OAAO,K;;MAEX,OAAO,0B;K;IAGX,g C;MAUI,IAAI,CAAQ,kBAAK,GAAL,CAAR,6BAAJ,C;QACI,OAAO,I;;MAEX,IAAI,YAAO,GAAX,C;QACI,OA AO,K;;MAEX,OAAO,0B;K;IAGX,gC;MASI,IAAI,YAAO,GAAX,C;QACI,OAAO,K;;MAEX,OAAO,gCAAoD,y

D;K;IAG/D,iC;MAUI,OAAO,aAAQ,EAAR,IAAoB,CAAQ,mBAAU,GAAV,CAAR,6B;K;IAG/B,iC;MAMiD,kC;K;iFwGtPjD,yB;MAAA,+C;MAAA,4B;QAMuD,OAAK,UAAAL,SAAK,C;O;KAN5D,C;IAQA,gC;MAMiD,4B;M AAA,S;QAAgB,cAAA,SvG4LC,cuG5LD,EAAoB,MAApB,C;;MAAhB,W;K;IAEjD,6B;MAI0C,Q;MAAA,yDAA kB,kBAAkB,SAAlB,C;K;IAE5D,oC;MAKoD,Q;MAAA,yCAAA,KAAb,oBAAuB,kBAAkB,SAAlB,C;K;IAG3E,8 B;MAI4C,Q;MAAA,0DAAmB,kBAAkB,SAAlB,C;K;IAE/D,qC;MAKsD,Q;MAAA,0CAAc,KAAAd,oBAAwB,kB AAkB,SAAlB,C;K;IAE9E,0B;MAIwC,Q;MAAA,wDAAiB,kBAAkB,SAAlB,C;K;IAEzD,mC;MAKkD,Q;MAAA, wCAAY,KAAZ,oBAAsB,kBAAkB,SAAlB,C;K;IAExE,2B;MAI0C,Q;MAAA,yDAAkB,kBAAkB,SAAlB,C;K;IA E5D,oC;MAKoD,Q;MAAA,yCAAA,KAAb,oBAAuB,kBAAkB,SAAlB,C;K;IAE3E,6B;MAIyF,kBAA1C,CAA0,S; MACID,IAAO,QjHeD,WiHfC,CAAH,IAAc,CAAM,kBAApB,KjHeE,WiHf6B,KAAM,GAAN,IAAkB,kBAAjD,C AAJ,C;QACI,4B;MAFsC,OjHiBnC,W;K;6EiHZX,yB;MAAA,6C;MAAA,4B;QAKmD,0B;O;KALnD,C;IAOA,mC ;MAIgg,kBAA1C,CAA0,S;MAAR,OAcjD,EAAK,QjH2BgB,WiH3BhB,CAAH,IAAc,CAAM,kBAApB,KjH2Bm B,WiH3BY,KAAM,GAAN,IAAkB,kBAAjD,CAAF,CjH2BO,GAAqB,WAArB,GAA+B,I;K;yFiHxB1C,yB;MAA A,yD;MAAA,4B;QAK0D,gC;O;KAL1D,C;iFAOA,yB;MAAA,6C;MAAA,mC;QAO6D,OAAa,SAAR,SAAQ,EAA S,KAAT,C;O;KAP1E,C;iFASA,yB;MAAA,6C;MAAA,mC;QAO8D,OAAa,SAAR,SAAQ,EAAAS,KAAT,C;O;KAP 3E,C;IASA,sC;MAMqD,OAAA,SAAY,UAAS,WAAW,KAAX,CAAT,C;K;IAEjE,4B;MAAsC,QAAM,SvG4EsB,c uG5E5B,C;aACIC,K;aAAA,M;aAAA,M;UADkC,OACT,I;;UADS,OAE1B,K;;K;IAGZ,2B;MAKI,IAAI,EAU,CA AV,sBAAa,EAAb,CAAJ,C;QACI,MAAM,gCAAYB,WAAQ,KAAR,kCAAzB,C;;MAEV,OAAO,K;K;IAGX,8B;M AA2D,Q;MACvD,YAAQ,EAAR,IAAe,QAAQ,EAAvB,C;QAA8B,cAAO,E;WACrC,YAAQ,EAAR,IAAe,QAAQ, EAAvB,C;QAA8B,cAAO,EAAP,GAAa,EAAb,I;WAC9B,YAAQ,EAAR,IAAe,QAAQ,GAAvB,C;QAA8B,cAAO, EAAP,GAAa,EAAb,I;WAC9B,WAAO,GAAP,C;QAAmB,S;WACnB,YAAQ,KAAR,IAAoB,QAAQ,KAA5B,C;Q AAwC,cAAO,KAAP,GAAkB,EAAlB,I;WACxC,YAAQ,KAAR,IAAoB,QAAQ,KAA5B,C;QAAwC,cAAO,KAAP ,GAAkB,EAAlB,I;;QAC3B,sBAAL,IAAK,C;MjH9CN,a;MiHuCgD,OAQ/C,WAAJ,GAAiB,EAajB,GAAyB,E;K;I CIJG,2C;MAHpC,e;MAGqC,kB;MAHrC,iB;MAAA,uB;K;IAAA,kC;MAAA,qC;O;MAIL,qEACY,GADZ,C;MAE A,iEAIU,GAJV,C;K;;IAFA,+C;MAAA,wB;MAAA,uC;K;;IAEA,6C;MAAA,wB;MAAA,qC;K;;IANJ,8B;MAAA, mF;K;;IAAA,mC;MAAA,a;aAAA,a;UAAA,4C;aAAA,W;UAAA,0C;;UAAA,4D;;K;;IAawG,4B;MAAE,OAAA,E AAG,M;K;IAA7G,qC;MAAqE,iCAAA,EAAb,EAA0B,OAA1B,0BAAmC,cAAAnC,C;K;IAQIC,2B;MAAC,kB;K;;s CALpC,Y;MAKoC,iB;K;wCALpC,iB;MAAA,sBAKoC,qCALpC,C;K;oCAA,Y;MAAA,OAKoC,iDALpC,M;K;o CAAA,Y;MAAA,c;MAKoC,sD;MALpC,a;K;kCAA,iB;MAAA,2IAKoC,sCALpC,G;K;IAQA,gC;MAUsB,gB;M AAA,iF;MAAA,mB;QACX,MAAM,qCAA8B,8DAA9B,C;;MADb,kBAAkB,M;MAGIB,OAAO,wBAAY,IAAZ,C; K;IAiBe,iC;MA4PtB,6B;MANPA,eAcCoC,O;MACpC,eAcSd,QAAR,OAAQ,C;MACTd,uBAAoC,WAAO,OAAP,E AAwB,QAAR,OAAQ,EAAQ,IAAR,CAAxB,C;MACpC,6BAA2C,I;MAI3C,oCAAkD,I;K;0CAHID,Y;MACI,Q;M AAA,U;MAAA,gD;QAAA,a;;QAA8D,gBAAvC,WAAO,YAAP,EAAwB,QAAR,YAAQ,EAAQ,IAAR,CAAxB,C; QAA8C,6BIHkbnE,S;QkHIBF,SIHmBG,S;;MkHnBH,a;K;iDAGJ,Y;MACI,Q;MAAA,U;MAAA,uD;QAAA,a;;QIH 3BG,gB;QkH4BC,IAAY,aAAR,YAAQ,EAAW,EAAX,CAAR,IAAmC,WAAO,YAAQ,EAAAS,EAAT,CAAvC,C;U AAA,eACI,oB;;UAEA,OAAO,WAAO,MAA2B,UAAf,YAAR,YAAQ,qBAAU,EAAV,EAAe,qBAAQ,EAAR,EAA 3B,MAAP,EAA2D,QAAR,YAAQ,EAAQ,IAAR,CAA3D,C;QACb,4B;QAAO,oCIHSP,S;QkHdF,SIHeG,S;;MkHf H,a;K;sCAQJ,iB;MAEkB,MAAd,oBAAc,C;MACd,YAAY,oBAAc,MAAK,KAAM,WAAAX,C;MAC1B,OAAO,iB AAiB,KAAM,MAAN,KAAe,CAAhC,IAAqC,oBAAc,UAAAd,KAA2B,KAAM,O;K;8CAGjF,iB;MAEkB,MAAd,oB AAc,C;MACd,OAAO,oBAAc,MAAK,KAAM,WAAAX,C;K;wCAGzB,wB;MAGI,IAAI,QAAQ,CAAR,IAAa,QAA Q,KAAM,OAA/B,C;QACI,MAAM,8BAA0B,0BAAuB,KAAvB,wBAA8C,KAAM,OAA9E,C;;MAEV,cAAc,0B; MACd,oBAAoB,K;MACpB,OAAO,OAAQ,MAAK,KAAM,WAAAX,C;K;mCAGnB,6B;MAS4C,0B;QAAA,aAAk B,C;MAC1D,IAAI,aAAa,CAAb,IAAkB,aAAa,KAAM,OAAzC,C;QACI,MAAM,8BAA0B,gCAA6B,UAA7B,wB AAyD,KAAM,OAAzF,C;;MAEV,OAAqB,SAAd,oBAAc,EAAS,KAAM,WAAf,EAA2B,UAA3B,EAAuC,oBAAv C,C;K;IAeG,6E;MAAA,mB;QAAE,+BAAK,aAAL,EAAY,kBAAZ,C;O;K;IAA2B,uC;MAAW,OAAA,KAAM,O; K;sCAZ1E,6B;MAQ+C,0B;QAAA,aAAkB,C;MAC7D,IAAI,aAAa,CAAb,IAAkB,aAAa,KAAM,OAAzC,C;QACI, MAAM,8BAA0B,gCAA6B,UAA7B,wBAAyD,KAAM,OAAzF,C;;MAEV,OAAO,mBAAiB,6CAAjB,EAA8C,sB AA9C,C;K;0CAGX,iB;MAMI,OAA2B,SAA3B,iCAA2B,EAAS,KAAM,WAAf,EAA2B,CAA3B,EAA8B,oBAA9 B,C;K;sCAE/B,wB;MAGI,IAAI,QAAQ,CAAR,IAAa,QAAQ,KAAM,OAA/B,C;QACI,MAAM,8BAA0B,0BAAuB

,KAAvB,wBAA8C,KAAM,OAA9E,C;;MAEV,OAA2B,SAApB,0BAAoB,EAAS,KAAM,WAAf,EAA2B,KAA3B,EAakC,oBAAIC,C;K;IA2BL,mD;MAAA,qB;QAAE,2BAAoB,EAAPB,EAAwB,mBAAxB,C;O;K;sCAvB5B,8B;MAoBI,IAAI,CAAA,YAAZ,WAAy,EAAS,EAAT,CAAb,IAA+B,CAAA,YAAZ,WAAy,EAAS,EAAT,CAAhD,C;QACI,OAAO,KAAM,WxGoF4E,SwGpFnD,oBxGoFmD,EwGpFpC,WxGoFoC,C;;MwGIF7F,OAAO,qBAAQ,KAAR,EAAe,iCAAf,C;K;sCAGX,4B;MAMI,YAAy,kBAAK,KAAL,C;MACZ,IAAI,aAAJ,C;QAAmB,OAAO,KAAM,W;MAEhC,gBAAGB,C;MACHB,aAAa,KAAM,O;MACnB,SAAS,mBAAC,MAAd,C;;QAEI,iBAAiB,oB;QACjB,EAAG,gBAAO,KAAP,EAAC,SAAd,EAAyB,UAAW,MAAM,MAA1C,C;QACH,EAAG,gBAAO,UAAU,UAAV,CAAP,C;QACH,YAAy,UAAW,MAAM,aAAjB,GAAgC,CAAhC,I;QACZ,QAAQ,UAAW,O;;MACd,oBAAy,MAAZ,IAAsB,aAAtB,C;MAET,IAAI,YAAy,MAAhB,C;QACI,EAAG,gBAAO,KAAP,EAAC,SAAd,EAAyB,MAAzB,C;;MAGP,OAAO,EAAG,W;K;2CAGd,8B;MAyBgB,Q;MALZ,IAAI,CAAA,YAAZ,WAAy,EAAS,EAAT,CAAb,IAA+B,CAAA,YAAZ,WAAy,EAAS,EAAT,CAAhD,C;QACI,uBAA+B,QAAR,YAAQ,EAAQ,GAAR,C;QAC/B,OAAO,KAAM,WxG8B4E,SwG9BnD,WAAO,YAAP,EAAGB,gBAAhB,CxG8BmD,EwG9BhB,WxG8BgB,C;;MwG3BjF,yBAAK,KAAL,C;MAAA,iB;QAAe,OAAO,KAAM,W;;MAAxC,YAAy,I;MCqKO,gBAAhB,sB;MDIKC,yBIG4Kf,0BkG5KzD,CIG4KyD,EkG5KhD,WAAAM,MIG4K0C,CAAkC,WkG5KIh,C;MACA,yBAAO,uCAAP,C;MACA,yBIG0Kf,0BkG1KnD,WAAAM,KAAs,GAAMB,CAANB,IIG0KyD,EkG1K7B,YIG0K6B,CAAkC,WkG1KIh,C;MAHJ,OlHjKJ,SmHoUqC,W;K;oCD5J5C,wB;MAO6C,qB;QAAA,QAAa,C;MAMxC,Q;MALd,wBAAwB,KAAxB,C;MIHjJG,SkHqJW,qBAAQ,KAAR,C;MAAd,cAAuC,UAAS,CAAb,GAAgB,EAAhB,GAA2B,OAAH,EAAG,EAAK,QAAQ,CAAR,IAAL,C;MAC9D,a/H1KgD,gB;M+H2KhD,gBAAGB,C;MAEF,yB;MAAd,OAAc,cAAAd,C;QAAC,uB;QACV,MAAO,WAAU,mBAAN,KAAM,EAAY,SAAZ,EAAuB,KAAM,MAAM,MAAnC,CAA0C,WAApD,C;QACP,YAAy,KAAM,MAAM,aAAZ,GAA2B,CAA3B,I;;MAEhB,MAAO,WAAU,mBAAN,KAAM,EAAY,SAAZ,EAAuB,KAAM,OAA7B,CAAqC,WAA/C,C;MACP,OAAO,M;K;IAgBS,yI;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,0C;MAAA,oC;MAAA,0C;MAAA,yB;MAAA,6B;MAAA,8B;MAAA,8B;MAAA,kC;K;;;gEAAA,Y;;;;iCACA,mCAAK,wBAAL,C;CACZ,IAAI,4BAAiB,6BAAS,CAA9B,C;gBACI,gB;gCAAA,iCAAM,wBAAAM,WAAZ,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBADJ,gB;;;;cAEI,M;;qCAGY,C;sCACC,C;cAEjB,gB;;;sCACqB,+B;cACjB,gB;8BAAA,iClGwH4E,mBkGxHtE,wBlGwHsE,EkGxHtD,oBlGwHsD,EkGxH3C,qBAAW,MAAM,MIGwH0B,CAAkC,WkGxH9G,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cACA,uBAAy,qBAAW,MAAM,aAAjB,GAAgC,CAAhC,I;cACZ,mBAAQ,qBAAW,O;cAJvB,KAKS,qDALT,EAKS,qBALT,OAKyB,2BAAQ,CAAR,IALzB,KAKsC,gBALtC,S;gBAAA,gB;;;cAAA,gB;;;cAOA,gB;8BAAA,iClGmHgF,mBkGnH1E,wBlGmH0E,EkGnH1D,oBlGmH0D,EkGnH/C,wBAAM,OIGmHyC,CAAkC,WkGnHIH,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cAhBA,OAgBA,a;;;;K;IAjBY,sF;MAAA,yD;uBAAA,6H;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;8CABpB,wB;MAUuD,qB;QAAA,QAAa,C;MACHE,wBAAwB,KAAxB,C;MAEA,OAAO,SAAS,gDAAT,C;K;+BASBX,Y;MAMyC,OAAA,oBAAC,W;K;IAEvD,2B;MAAA,+B;MAMBI,uBAA4B,WAAO,uBAAP,EAAiC,GAAjC,C;MAC5B,2BAAgC,WAAO,SAAP,EAAoB,GAAPB,C;MAGhC,iCAAsC,WAAO,KAAP,EAAiB,GAAjB,C;K;oDAAtBtC,mB;MAIwD,oBAAM,oBAAO,OAAP,CAAN,C;K;+CAExD,mB;MAIoD,OAAA,OxGzDyC,SwGyDnB,oBxGzDmB,EwGyDJ,MxGzDI,C;K;0DwG2D7F,mB;MAI+D,OAAA,OxG/D8B,SwG+DR,wBxG/DQ,EwG+DW,MxG/DX,C;K;gEwGoE7F,mB;MAAGe,OAAA,OxGpE6B,SwGoEP,8BxGpEO,EwGoEkB,MxGpElB,C;K;;IwG8CjG,uC;MAAA,sC;QAAA,qB;;MAAA,+B;K;;IA1PA,4C;MAAA,+C;MACkE,kBAAK,OAAL,EAAC,MAAM,MAAN,CAAd,C;MADIE,Y;K;IAGA,sC;MAAA,+C;MAC6C,kBAAK,OAAL,EAAC,UAAAd,C;MAD7C,Y;K;IA0RO,kG;MAAA,kC;MAAA,8C;MAAA,kC;MAAA,kC;MACH,uBAA+B,a;MAI/B,4F;MA0BA,sBAA0C,I;K;+FA9B1C,Y;MAAA,2B;K;+FAEI,Y;MAAQ,qBAAA,kBN9S8C,CM8SxC,CN9SwC,CM8S9C,C;K;gGAEZ,Y;MAAA,4B;K;iEAqBA,mB;MACI,OAAO,MAAA,UAAU,eAAe,MAAK,CAAL,EAAQ,IAAR,C;K;IAStB,oG;MAAA,kC;MAAS,uB;K;mJACG,Y;MAAQ,OAAA,kBAAM,O;K;wGACrC,iB;MAAuC,Q;MAAA,eAAA,kBNjVG,CMiVG,KNjVH,CMiVH,mBAAgB,E;K;;qGAJnE,Y;MACI,IAAI,2BAAJ,C;QACI,yH;;MAKJ,OAAO,kC;K;4CAGf,Y;MACI,OAAy,SAAZ,wBAAy,EAAS,kBAAT,EAAoB,kBAAM,UAAV,GAAqB,8BAAuB,kBAAM,MAA7B,CAArB,GAA8D,kBAAM,aAAN,GAAqB,CAArB,IAA9E,EAAsg,wBAAtG,C;K;gEAehB,iB;MACI,IAAI,QAAC,iBAAN,kBAAM,CAAIB,C;QACI,YAAkB,kBAAy,YAAW,KAAX,C;QAC9B,IAAa,KAAT,sBAAiB,KAAR,B,C;UACI,YAAkB,kBAAy,YAAW,QAAQ,CAAR,IAAX,C;UAC9B,IAAa,KAAT,sBAAiB,KAAR,B,C;YACI,OAAO,QAAQ,CAAR,I;;;MAInB,OAAO,QAAQ,CAAR,I;K;IApDiC,2E;MAAA,kC;MAAA,kB;MAAoC,6B;K;mHACrD,Y;MAAQ,OAAA,kBAAM,O;K;IACqC,4E;MA

AA,qB;QAAE,yBAAK,EAAL,C;O;K;qEAA5E,Y;MAAiD,OAAqB,OAAb,aAAR,oBAAQ,CAAa,EAAl,iEAAJ,C  
AAiB,W;K;wEACvF,iB;MAA4C,Q;MAAA,eAAA,kBNnTU,CMmTJ,KNnTI,CMmTV,YAAoB,oBAApB,O;K;wE  
AE5C,gB;MAGmC,UASqB,MATrB,EASxB,M;MATwB,OAAZ,kBAAY,O;MAAIB,iB;QACN,MAAM,gCAAyB,  
gCAA6B,IAA7B,oEAAzB,C;;MADb,aAAa,I;MAKb,IAAI,CAAC,qCAAwB,MAAxB,EAAGC,IAAhC,CAAL,C;Q  
ACI,MAAM,gCAAyB,gCAA6B,IAA7B,qBAAzB,C;MAEV,YAAy,OAAO,IAAP,C;MACL,IAAI,SAAS,SAAb,C;  
QAAwB,a;QAAU,wBAAW,4DAAX,C;;MAAzC,a;K;;;IA5BhB,uD;MACI,sBAAiB,I;MACjB,YAAy,eAAK,KAA  
L,C;MACZ,IAAI,aAAJ,C;QAAmB,OAAO,I;MAC1B,YAAy,aAAA,KAAM,MAAN,EAAa,sBAAY,CAAZ,IAAb,  
C;MAEZ,mE;K;IA8DJ,iD;MAM+B,UAKO,MALP,EAoBD,MApBC,EAoBD,MApBC,EAiCD,MAjCC,EAiCD,M;  
MArC1B,YAAy,C;MACZ,aAAa,sB;MAEb,OAAO,QAAQ,WAAy,OAA3B,C;QACI,WAAW,wBAAy,YAAZ,EA  
AY,oBAAZ,Q;QACX,IAAI,SAAQ,EAZ,C;UACI,IAAI,UAAS,WAAy,OAAzB,C;YACI,MAAM,gCAAyB,mCA  
AzB,C;UAEV,MAAO,gBAAO,wBAAy,cAAZ,EAAy,sBAAZ,UAAP,C;eACJ,IAAI,SAAQ,EAZ,C;UACH,IAAI  
,UAAS,WAAy,OAAzB,C;YACI,MAAM,gCAAyB,kCAAzB,C;UAEV,IAAI,uBAAy,KAAZ,MAAsB,GAA1B,C;  
YACI,eAA2B,cAAZ,WAAy,GAAC,qBAAd,EAAC,KAAd,E;YAE3B,IAAI,UAAS,QAAb,C;cACI,MAAM,gCAAy  
B,8DAAzB,C;YACV,IAAI,aAAy,WAAy,OAAxB,IAAkC,uBAAy,QAAZ,MAAyB,GAA/D,C;cACI,MAAM,gC  
AAyB,yDAAzB,C;YAEV,gBAAgB,WxGvLgE,WwGuL1C,KxGvL0C,EwGuLnC,QxGvLmC,C;YwGyLhF,MAA  
O,gBAAO,0BAAA,KAAM,OAAN,EAAa,SAAb,qDAaKc,EAazC,C;YACP,QAAQ,WAAW,CAAX,I;;YAER,IA  
AI,EAAuB,kBAAK,EAAL,CAAvB,0CAAy,KAAZ,EAAJ,C;cACI,MAAM,gCAAyB,mCAAzB,C;YAEV,aAAa,K  
AAM,O;YACnB,iBAA2B,eAAZ,WAAy,EAae,KAAf,EAAsB,MAAO,KAA7B,C;YAC3B,iBAAwD,MAAvC,Wx  
GjM+D,WwGiMzC,KxGjMyC,EwGiMIC,UxGjMkC,CwGiMxB,C;YAExD,IAAI,cAAc,MAAO,KAAzB,C;cACI,  
MAAM,8BAA0B,sBAAmB,UAAAnB,oBAA1B,C;YAEV,MAAO,gBAAO,uCAAo,UAAp,qDAA6B,EAAPC,C;YA  
CP,QAAQ,U;;;UAGZ,MAAO,gBAAO,IAAP,C;;;MAGf,OAAO,MAAO,W;K;IAG1B,8C;MAKI,YAAy,U;MACZ,  
OAAO,QAAQ,gBAaf,C;QACI,IAAI,qBAAK,KAAL,MAAE,GAAnB,C;UACI,K;;UAEA,qB;;MAGR,OAAO,K;K  
;IAGX,2D;MAEI,YAAy,aAAa,CAAb,I;MACZ,iBAAiB,qBAAK,UAAL,IAAmB,E;MAGpC,OAAO,QAAQ,gBA  
AR,IAAkB,CAAE,kBAAK,EAAL,CAAF,wCAAK,KAAL,EAazB,C;QACI,oBAAoB,CAAC,aAAa,EAAb,IAAD,K  
AAqB,qBAAK,KAAL,IAAc,EAAnC,K;QACpB,IAAqB,CAAjB,qCAAyB,UAA7B,C;UACI,aAAa,a;UACb,qB;;U  
AEA,K;;MAGR,OAAO,K;K;IxGneX,yB;MAQiB,Q;MADb,aAAa,E;MACb,wBAAa,KAAb,gB;QAAa,WAAb,UA  
Aa,KAAb,O;QACI,8BAAU,IAAV,C;;MAEJ,OAAO,M;K;IAGX,yC;MAA+B,Q;MAH3B,IAAI,SAAS,CAAT,IAAc  
,SAAS,CAAvB,IAA4B,CAAA,KAAM,OAAN,GAAa,MAAb,QAAAsB,MAAtD,C;QACI,MAAM,8BAA0B,WAAS,  
KAAM,OAAf,kBAA+B,MAA/B,kBAAgD,MAA1E,C;MACV,aAAa,E;MACc,gBAAS,MAAT,I;MAA3B,iBAAc,  
MAAd,wB;QACI,8BAAU,MAAM,KAAN,CAAV,C;;MAEJ,OAAO,M;K;IAGX,mC;MAOiB,Q;MADb,aAAa,E;M  
ACb,wBAAa,SAAb,gB;QAAa,WAAb,UAAa,SAAb,O;QACI,8BAAU,IAAV,C;;MAEJ,OAAO,M;K;IAGX,2D;MA  
Y2C,0B;QAAA,aAAkB,C;MAAG,wB;QAAA,WAAgB,SAAK,O;MACjF,oCAAa,4BAAmB,UAAAnB,EAA+B,QA  
A/B,EAAyC,SAAK,OAA9C,C;MACb,aAAa,E;MACb,iBAAc,UAAAd,UAA+B,QAA/B,U;QACI,8BAAU,UAAK,K  
AAL,CAAV,C;;MAEJ,OAAO,M;K;IASkB,gD;MAAA,qB;QAAE,+CAAL,EAAJ,E;O;K;IAN/B,kC;MAMI,OAAO,  
kBAAU,gBAAV,EAakB,+BAAIB,C;K;IAiBiC,oE;MAAA,qB;QAAE,+CAAL,qBAAa,EAAb,IAAJ,E;O;K;IA9C,  
wD;MAYqC,0B;QAAA,aAAkB,C;MAAG,wB;QAAA,WAAgB,SAAK,O;MAC3E,oCAAa,4BAAmB,UAAAnB,EA  
A+B,QAA/B,EAAyC,gBAAzC,C;MACb,OAAO,kBAAU,WAAW,UAAX,IAAV,EAAC,2CAAjC,C;K;IAGX,mC;  
MAQI,OAAO,WAAW,SAAX,EAAiB,CAAjB,EAAoB,gBAApB,EAA0B,KAA1B,C;K;IAGX,mF;MAeI,0B;QAA  
A,aAAkB,C;MACIB,wB;QAAA,WAAgB,SAAK,O;MACrB,sC;QAAA,yBAakC,K;MAEIC,oCAAa,4BAAmB,UA  
AnB,EAA+B,QAA/B,EAAyC,SAAK,OAA9C,C;MACb,OAAO,WAAW,SAAX,EAAiB,UAAjB,EAA6B,QAA7B,  
EAAuC,sBAAvC,C;K;IAGX,sC;MAQI,OAAO,WAAW,SAAX,EAAiB,CAAjB,EAAoB,gBAApB,EAA4B,KAA5  
B,C;K;IAGX,sF;MAeI,0B;QAAA,aAAkB,C;MACIB,wB;QAAA,WAAgB,SAAK,O;MACrB,sC;QAAA,yBAakC,  
K;MAEIC,oCAAa,4BAAmB,UAAAnB,EAA+B,QAA/B,EAAyC,gBAAzC,C;MACb,OAAO,WAAW,SAAX,EAAiB  
,UAAjB,EAA6B,QAA7B,EAAuC,sBAAvC,C;K;uFAGX,qB;MAMwD,OAAA,SAAY,c;K;mFAEpE,qB;MAWsD,  
OAAA,SAAY,c;K;uFAEIE,qB;MAMwD,OAAA,SAAY,c;K;mFAEpE,qB;MAWsD,OAAA,SAAY,c;K;yFAEIE,qC  
;MACoF,OAAA,SAAY,SAAQ,GAAR,EAAa,SAAb,C;K;IGAEhG,qC;MACwF,OAAA,SAAY,aAAy,GAZ,EA  
iB,SAAjB,C;K;+FAEpG,kC;MAWiF,OAAA,SAAY,YAAW,CAAX,EAAC,QAAAd,C;K;2FAE7F,wB;MAGBgE,OA  
AA,SAAY,UAAS,CAAT,C;K;iFAE5E,iC;MACqE,OAAA,SAAY,WAAU,UAAV,C;K;mFAEjF,2C;MACoF,OAA

A,SAAY,WAAU,UAAV,EAAaB,QAAtB,C;K;4EAehG,0B;MAGuD,OAAA,SAAY,QAAO,GAAP,C;K;wEAEnE,4B;MAGgE,OAAA,SAAY,OAAM,KAAAN,C;K;yFAK5E,2C;MACyF,OAAA,SAAY,SAAQ,OAAR,EAAiB,WAAjB,C;K;IAErG,iD;MAOkD,0B;QAAA,aAAsB,K;MACpE,IAAI,UAAJ,C;QACI,SAAS,SAAK,O;QACd,SAAS,KAA M,O;QACf,UtTBG,MAAO,KSsBM,ETtBN,ESsBU,ETtBV,C;QSuBV,IAAI,QAAO,CAAX,C;UAAc,OAAO,KAA K,EAAL,I;QACrB,iBAAc,CAAd,UAAaB,GAAtB,U;UACI,eAAe,qBAAK,KAAL,C;UACf,gBAAGB,iBAAM,KAA AN,C;UAEhB,IAAI,aAAY,SAAhB,C;YACI,WAAoB,cAAT,QAAS,C;YACpB,YAAaB,cAAV,SAAU,C;YAEtB,IA AI,aAAY,SAAhB,C;cACwB,kBAAT,Q;cAAX,WD3P2C,gCAAY,cAfrB,YAAY,CAAZ,C;c2QZ,kBAAV,S;cA AZ,YD5P2C,gCAAY,cAfrB,YAAY,CAAZ,C;c6Q1C,IAAI,aAAY,SAAhB,C;gBACI,OAAgB,iBAAT,QAAS,EA AU,SAAV,C;QAKhC,OAAO,KAAK,EAAL,I;QAEP,OAAO,4BAAU,KAAV,C;K;IAIf,4C;MAOqF,oCAAkB, KAAIB,C;K;IAErF,wD;MASI,OAAW,UAAJ,GACE,4BAAL,SAAK,EAA4B,KAA5B,CADF,GAGE,kBAAL,SA A K,EAakB,KAAIB,C;K;IAIkD,oD;MAAU,OAAE,UAAF,CAAE,EAAU,CAAV,EAA0B,IAA1B,C;K;IAIvE,+C;M AAQ,oC;K;2F0GxUZ,oC;MACiF,O1G2Me,kB0G3ME,oBAAH,EAAG,C1G2MF,E0G3Mc,S1G2Md,C;K;mG0Gz MhG,oC;MACqF,O1G2Me,sB0G3MM,oBAAH,EAAG,C1G2MN,E0G3MkB,S1G2MIB,C;K;I0GzMpG,mD;MAIo D,0B;QAAA,aAAsB,K;MACtE,IAAI,CAAC,UAAAL,C;QACI,O1GgNqF,qB0GhN7D,M1GgN6D,E0GhNrD,C1Gg NqD,C;Q0G9MrF,OAAO,yBAAc,CAAd,EAAiB,MAAjB,EAAyB,CAAzB,EAA4B,MAAO,OAAnc,EAA2C,UA A3C,C;K;IAGf,iE;MAIqE,0B;QAAA,aAAsB,K;MACvF,IAAI,CAAC,UAAAL,C;QACI,O1GqMqF,qB0GrM7D,M1 GqM6D,E0GrMrD,U1GqMqD,C;Q0GnMrF,OAAO,yBAAc,UAAAd,EAA0B,MAA1B,EAakC,CAAIC,EAAqC,M AAO,OAA5C,EAAoD,UAApD,C;K;IAGf,iD;MAIkD,0B;QAAA,aAAsB,K;MACpE,IAAI,CAAC,UAAAL,C;QACI, O1G4MoE,mB0G5M9C,M1G4M8C,C;Q0G1MpE,OAAO,yBAAc,mBAAS,MAAO,OAAhB,IAAd,EAAaC,MAAt C,EAA8C,CAA9C,EAAiD,MAAO,OAAxD,EAAG,EAAhE,C;K;IAGf,mC;MAGI,aACa,S1GmN2D,O0GnNhD,K 1GmNgD,C;M0GInxE,OAAO,kBAakB,MAAO,OAAP,KAAe,C;K;IAG5C,4B;MAKoD,gCAAU,C;MAAV,U;QA AuB,kBAAR,yB;QAAQ,c;UjH6nDvD,U;UADhB,IAAI,0CAAsB,qBAA1B,C;YAAqC,aAAO,I;YAAP,e;UACrB, +B;UAAhB,OAAgB,gBAAhB,C;YAAgB,2B;YAAM,IAAI,CiH7nD4D,aAAT,qBjH6nDxC,OiH7nDwC,CAAS,Cj H6nDhE,C;cAAyB,aAAO,K;cAAP,e;UAC/C,aAAO,I;Qih9nDgE,iB;MAAvB,W;K;IAEpD,gD;MASiD,0B;QA AA,aAAsB,K;MAOxC,Q;MAN3B,IAAI,iBAAJ,C;QAAkB,OAAO,a;MACzB,IAAI,aAAJ,C;QAaMB,OAAO,K;M AC1B,IAAI,CAAC,UAAAL,C;QAAiB,OAAO,kBAAQ,KAAAR,C;MAExB,IAAI,SAAK,OAAL,KAAe,KAAAM,OAA zB,C;QAAiC,OAAO,K;MAEb,OAAL,SAAK,O;MAA3B,iBAAc,CAAd,wB;QACI,eAAe,qBAAK,KAAL,C;QACf, gBAAGB,iBAAM,KAAAN,C;QACHB,IAAI,CAAU,SAAT,QAAS,EAAO,SAAP,EAakB,UAAIB,CAAd,C;UACI,O AAO,K;MAIf,OAAO,I;K;IAIX,sF;MACkH,0B;QAAA,aAAsB,K;MACpI,oCAAkB,UAAIB,EAA8B,KAA9B,EA AqC,WAArC,EAakD,MAAID,EAA0D,UAA1D,C;K;IAGJ,+B;MAYI,OpGmMmD,mBAAS,CoGnM5D,G1GiJ4F, oB0GjzD,C1GiJyD,E0GjJtD,C1GiJsD,CAhE9B,c0GjFrC,G1G8IoD,oB0G9IZ,C1G8IY,C0G9I7E,GAAyE,S;K;IA G7E,iC;MASI,OpGuLmD,mBAAS,CoGvL5D,G1GqI4F,oB0GrIzD,C1GqIyD,E0GrItD,C1GqIsD,CA3C9B,c0G1Fr C,G1GkIoD,oB0GIIZ,C1GkiY,C0GI7E,GAAyE,S;K;IAG7E,8B;MAOiB,IAAN,I;MvH/FP,IAAI,EUH8FI,KAAK, CvH9FT,CAAJ,C;QACI,cuH6Fc,oD;QvH5Fd,MAAM,gCAAYB,OAAQ,WAAJ,C;MuH6FH,QAAM,CAAN,C;a ACH,C;UAAK,S;UAAAL,K;aACA,C;UAAU,OAAL,SAAK,W;UAAV,K;UAEI,aAAa,E;UACb,IAAI,EpGgKoC,qB AAU,CoGhK9C,CAAJ,C;YACI,QAAQ,SAAK,W;YACb,YAAY,C;YACZ,OAAO,IAAP,C;cACI,IAAI,CAAC,QA AU,CAAX,MAAiB,CAArB,C;gBACI,UAAU,C;cAEd,QAAQ,UAAW,C;cACnB,IAAI,UAAAS,CAAb,C;gBACI,K; ;cAEJ,KAAK,C;UAGb,OAAO,M;MANbF,W;K;IAwBJ,4D;MAOqE,0B;QAAA,aAAsB,K;MACvF,O1G2GiG,k B0G3GnF,WAAO,6BAAM,gBAAO,QAAP,CAAb,EAAMC,UAAJ,GAAgB,KAAhB,GAA2B,IAA1D,C1G2GmF, E0G3GIB,6BAAM,iCAAwB,QAAXB,C1G2GY,C;K;I0GzGrG,4D;MAM+D,0B;QAAA,aAAsB,K;MACjF,O1GkGi G,kB0GIgnF,WAAO,6BAAM,gBAAe,oBAAR,OAAQ,CAAF,CAAb,EAA6C,UAAJ,GAAgB,KAAhB,GAA2B,IA ApE,C1GkGmF,E0GIGA,oBAAR,OAAQ,C1GkGA,C;K;I0GhGrG,iE;MAC0E,0B;QAAA,aAAsB,K;MAC5F,O1G 8FiG,kB0G9FnF,WAAO,6BAAM,gBAAO,QAAP,CAAb,EAAMC,UAAJ,GAAgB,IAAhB,GAA0B,GAAzD,C1G8 FmF,E0G9FpB,6BAAM,iCAAwB,QAAXB,C1G8Ff,C;K;I0G5FrG,iE;MACoE,0B;QAAA,aAAsB,K;MACiF,O1G0 FiG,kB0G1FnF,WAAO,6BAAM,gBAAe,oBAAR,OAAQ,CAAF,CAAb,EAA6C,UAAJ,GAAgB,IAAhB,GAA0B,G AAnE,C1G0FmF,E0G1FF,oBAAR,OAAQ,C1G0FE,C;K;I2GtQrG,kD;MAEL,IAAI,gBAAJ,C;QAAsB,MAAM,6B AAYB,qCAAKC,QAAQ,CAAR,IAAIC,CAAzB,C;MAC5B,OAAO,CAAC,IAAD,I;K;IAGX,iF;MAQI,IAAI,EAAS, KAAT,oBAAiB,KAAjB,KAA2B,SAAS,QAAXC,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,

C;;MAEX,UAAU,kBAAO,KAAP,CzGyBgC,I;MyGxB1C,IAAI,EAAQ,KAAR,kBAAGB,KAAhB,CAAJ,C;QACI, OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAAPB,C;;MAEX,OAAO,SAAW,CAAC,OAAS,IAAV,KAAqB,E AAhC,IAAwC,MAAQ,I;K;IAG3D,yE;MAQI,IAAI,SAAU,EAAV,MAAkB,CAAIB,IAAuB,SAAS,QAApC,C;QA CI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAAPB,C;;MAEX,YAAY,KAAa,CAAP,KAAO,C;MACzB,IAA I,SAAU,GA AV,MAAkB,GAAtB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAAPB,C;;MAEX,OAA Q,SAAU,CAAX,GAakB,KAAIB,GAA4B,I;K;IAGvC,yE;MASI,IAAI,SAAS,QAAb,C;QACI,OAAO,UAAU,CAA V,EAAa,KAAb,EAAoB,gBAAPB,C;;MAGX,YAAY,KAAa,CAAP,KAAO,C;MACzB,IAAI,SAAU,EAAV,MAAi B,CAArB,C;QACI,IAAI,SAAU,GA AV,MAAkB,GAAtB,C;UAEI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gB AAPB,C;;aAER,IAAI,SAAU,EAAV,MAAiB,EAARb,C;QACH,IAAI,SAAU,GA AV,MAAkB,GAAtB,C;UAEI,OA AO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAAPB,C;;aAER,IAAI,SAAU,GA AV,MAAkB,GAAtB,C;QACH,OAA O,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAAPB,C;;MAGX,IAAI,SAAQ,CAAR,UAAa,QAAjB,C;QACI,OAAO, UAAU,CAAV,EAAa,KAAb,EAAoB,gBAAPB,C;;MAEX,YAAY,KAAiB,CAAX,QAAQ,CAAR,IAAW,C;MAC7 B,IAAI,SAAU,GA AV,MAAkB,GAAtB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAAPB,C;;MAG X,OAAQ,SAAU,EAAX,GAAoB,SAAU,CAA9B,GAAqC,KAArC,GAA+C,O;K;IAG1D,yE;MASI,IAAI,SAAS,Q AA b,C;QACI,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAAPB,C;;MAGJ,YAAY,KAAa,CAAP,KAAO,C;MACzB,I AAI,SAAU,EAAV,MAAiB,CAArB,C;QACI,IAAI,SAAU,GA AV,KAakB,GAAtB,C;UAEI,OAAO,UAAU,CAAV ,EAAa,KAAb,EAAoB,gBAAPB,C;;aAER,IAAI,SAAU,EAAV,MAAiB,CAArB,C;QACH,IAAI,SAAU,GA AV,MA AkB,GAAtB,C;UAEI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAAPB,C;;aAER,IAAI,SAAU,EAAV,IAAg B,CAApB,C;QACH,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAAPB,C;aACJ,IAAI,SAAU,GA AV,MAAkB, GAAtB,C;QACH,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAAPB,C;;MAGX,IAAI,SAAQ,CAAR,UAAa,Q AAjB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAAPB,C;;MAEX,YAAY,KAAiB,CAAX,QAAQ,C AAR,IAAW,C;MAC7B,IAAI,SAAU,GA AV,MAAkB,GAAtB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAo B,gBAAPB,C;;MAGX,IAAI,SAAQ,CAAR,UAAa,QAAjB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,g BAApB,C;;MAEX,YAAY,KAAiB,CAAX,QAAQ,CAAR,IAAW,C;MAC7B,IAAI,SAAU,GA AV,MAAkB,GAAtB ,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAAPB,C;;MAEX,OAAQ,SAAU,EAAX,GAAoB,SAAU, EAA9B,GAAuC,SAAU,CAAjD,GAAwD,KAAxD,GAakE,O;K;;IAmb7E,oE;MAkB0B,UAGJ,MAHI,EAKJ,MA LLEAMJ,MANI,EASJ,MATI,EAUJ,MAVI,EA WJ,MAXI,EA gBA,MAhBA,EAiBA,MAjBA,EAkBA,MAiBA,EAo BA,MApBA,EAqBA,OA rBA,EAsBA,OA tBA,EAuBA,O;MxH9JtB,IAAI,EwHgII,cAAc,CAA d,IAAmB,YAAY,M AAO,OAAtC,IAAgD,cAAc,QxHhIIE,CAAJ,C;QACI,cAda,qB;QAeb,MAAM,gCAAYB,OAAQ,WAAjC,C;;MwHg IV,YAAY,cAAU,CAAC,WAAW,UAA X,IAAD,IAA0B,CAA1B,IAAV,C;MACZ,gBAAGB,C;MACHB,gBAAGB,U ;MAEhB,OAAO,YAAY,QAA nB,C;QACI,WAAW,mBAAO,gBAAP,EAAO,wBAAP,QzGzH2B,I;QyG2HIC,WAA O,GAAP,C;UACI,MAAM,kBAAN,EAAM,0BAAN,YAA0B,OAAL,IAAK,C;eAC9B,WAAO,IAAP,C;UACI,MA AM,kBAAN,EAAM,0BAAN,YAA4C,OAARb,QAAS,CAAV,GAAGB,GAAM,C;UAC5C,MAAM,kBAAN,EAAM ,0BAAN,YAA+C,OAAXB,OAAS,EAAV,GAAMB,GAAM,C;eAEnD,WAAO,KAAP,IAAiB,QAAQ,KAAzB,C;UA CI,MAAM,kBAAN,EAAM,0BAAN,YAA6C,OAAtB,QAAS,EAAV,GAAiB,GAAM,C;UAC7C,MAAM,kBAAN, EAAM,0BAAN,YAAuD,OAA/B,QAAS,CAAV,GAAiB,EAiB,GAA2B,GAAM,C;UACvD,MAAM,kBAAN,EA AM,0BAAN,YAA+C,OAAXB,OAAS,EAAV,GAAMB,GAAM,C;;UAG/C,gBAAGB,uBAAuB,MAAvB,EAA+B,I AA/B,EAAqC,SAArC,EAAgD,QAahD,EAA0D,gBAA1D,C;UACHB,IAAI,aAAa,CAAJB,C;YACI,MAAM,kBAA N,EAAM,0BAAN,YAAqB,0BAA0B,CAA1B,C;YACrB,MAAM,kBAAN,EAAM,0BAAN,YAAqB,0BAA0B,CAA 1B,C;YACrB,MAAM,kBAAN,EAAM,0BAAN,YAAqB,0BAA0B,CAA1B,C;;YAErB,MAAM,kBAAN,EAAM,0B AAN,YAAkD,OAA3B,aAAc,EAAf,GAASB,GAAM,C;YACID,MAAM,mBAAN,EAAM,2BAAN,aAA6D,OAARc ,aAAc,EAAf,GA AuB,EAAXB,GA AiC,GAAM,C;YAC7D,MAAM,mBAAN,EAAM,2BAAN,aAA4D,OAAPc,aAA c,CAAF,GAASB,EAAvB,GAAGC,GAAM,C;YAC5D,MAAM,mBAAN,EAAM,2BAAN,aAAoD,OAA7B,YAAc,E AAF,GA AwB,GAAM,C;YACpD,6B;;;MAMhB,OAAW,KAAM,OAAN,KAAc,SAAlB,GAA6B,KAA7B,GAA8C, UAAN,KAAM,EAAO,SAAP,C;K;;IAQzD,mE;MAiByB,Q;MxH9LrB,IAAI,EwHwLI,cAAc,CAA d,IAAmB,YAA Y,KAAM,OAARc,IAA6C,cAAc,QxHxL/D,CAAJ,C;QACI,cAda,qB;QAeb,MAAM,gCAAYB,OAAQ,WAAjC,C;; MwHwLV,gBAAGB,U;MACHB,oBAAoB,sB;MAEpB,OAAO,YAAY,QAA nB,C;QACI,WAAW,KAAmB,CAAb,g BAAa,EAAb,wBAAa,O;QAE1B,YAAQ,CAAR,C;UACI,aAAc,gBAAY,OAAL,IAAK,CAAZ,C;aACIB,YAAS,CA

AT,KAAc,EAAd,C;UACI,WAAW,eAAe,KAAf,EAAsB,IAAtB,EAA4B,SAA5B,EAAuC,QAAvC,EAAd,C;UACX,IAAI,QAAQ,CAAZ,C;YACI,aAAc,gBAAO,gBAAP,C;YACd,yBAAa,CAAC,IAAD,IAAb,K;;YAEA,aAAc,gBAAY,OAAL,IAAK,CAAZ,C;YACd,wBAAa,CAAb,I;;eAGR,YAAS,CAAT,KAAc,EAAd,C;UACI,aAAW,eAAe,KAAf,EAAsB,IAAtB,EAA4B,SAA5B,EAAuC,QAAvC,EAAd,C;UACX,IAAI,UAAQ,CAAZ,C;YACI,aAAc,gBAAO,gBAAP,C;YACd,yBAAa,CAAC,MAAD,IAAb,K;;YAEA,aAAc,gBAAY,OAAL,MAAK,CAAZ,C;YACd,wBAAa,CAAb,I;;eAGR,YAAS,CAAT,KAAc,EAAd,C;UACI,aAAW,eAAe,KAAf,EAAsB,IAAtB,EAA4B,SAA5B,EAAuC,QAAvC,EAAd,C;UACX,IAAI,UAAQ,CAAZ,C;YACI,aAAc,gBAAO,gBAAP,C;YACd,yBAAa,CAAC,MAAD,IAAb,K;;YAEA,WAAW,MAAD,GAAQ,KAAR,IAAqB,EAARb,GAA2B,K;YACtC,UAAW,SAAS,IAAV,GAAoB,K;YAC9B,aAAc,gBAAY,OAAL,IAAK,CAAZ,C;YACd,aAAc,gBAAW,OAAJ,GAAI,CAAX,C;YACd,wBAAa,CAAb,I;;UAIJ,UAAU,CAAV,EAAa,SAAb,EAAWb,gBAAxB,C;UACA,aAAc,gBAAO,gBAAP,C;;MAK1B,OAAO,aAAc,W;K;ICtQzB,uC;MAU2D,OAAwB,CAAXB,2BAAwB,mBAAS,SAAT,C;K;IAEnF,oC;MAKI,OAAQ,OAAW,mBAAL,SAAK,CAAX,C;K;IAGZ,6C;MAMI,IAAI,cAAS,SAAb,C;QACI,iBAAsB,SAAY,Y;QACIC,IAAI,kBAAJ,C;UACS,SAAL,eAA+B,iBAAc,SAAd,E;;UAE/B,UAAW,WAAL,SAAJ,C;;K;IAUnB,6C;MAC4B,UAAjB,M;MAAP,OAAO,WAAiB,OAAZ,SAAY,YAAjB,4CAA+D,W;K;IAI9E,iC;MACI,gBAAqB,sB;MACrB,iBAAsB,E;MACtB,kBAA+B,E;MAC/B,uBAAiC,C;K;uDAEjC,qB;MACc,qBAAV,SAAU,EAAC,EAAd,EAakB,EAAlB,C;MACV,OAAO,aAAO,W;K;gDAGIB,qB;MAA6D,gBAAR,c;MAAQ,c;;QxIm2Y7C,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UAAAsB,IAAc,OwIn2Y+B,cxIm2Y7C,C;YAAwB,aAAO,I;YAAP,e;;QAC9C,aAAO,K;;MwIp2Y8C,iB;K;sDAErD,wC;MACI,KAAK,qBAAL,SAAK,EAAC,MAAd,EAAsB,SAATB,CAAL,C;QAAyC,M;MAEzC,YAAY,SAAK,M;MACjB,OAAO,aAAP,C;QACI,KAAM,qBAAN,KAAAM,EAAC,MAAd,EAAsB,aAAtB,CAAN,C;UAA8C,M;QAC9C,QAAQ,KAAM,M;;K;sDAItB,wC;MASgB,IAAiB,IAAjB,EA2BE,M;MAnCd,aAAO,gBAAO,MAAP,CAAe,gBAAO,SAAP,C;MACtB,gBAAgB,SAAK,W;MACrB,IAAI,eAAQ,SAAR,CAAJ,C;QACI,aAAO,gBAAO,kCAAP,CAA2C,gBAAO,SAAP,CAAKB,gBAAO,KAAP,C;QACpE,OAAO,K;;MAEH,cAAY,MAAK,SAAL,C;MAEpB,YAAY,CAAiB,OAAZ,SAAY,MAAjB,2D;MACZ,IAAI,aAAJ,C;QtHyBG,SsHxBwB,WAAN,KAAM,EAQ,SAAR,C;QAAvB,iBAAoD,KAAK,CAAT,GAAY,CAAZ,GAAmB,KAAe,gBAAf,I;QACnE,IAAI,eAAc,CAAlB,C;UAAqB,aAAO,gBAAO,SAAP,CAAKB,gBAAO,IAAP,C;QAC9C,IAAI,etG8MoC,YAAU,CsG9MID,C;UACI,kBAAW,K;UACX,uBAAgB,U;;UAEhB,QAAQ,wBAAiB,KAAjB,EAAwB,UAAxB,C;;QAEZ,IAAI,MtGgNuC,UAAS,CsGhNpD,C;UAEuB,U;UAAA,IAAI,eAAc,CAAlB,C;YAAA,SAAQB,C;;YvGs+BpC,U;YADhB,YAAY,C;YACI,oBuGt+B+C,SvGs+B/C,C;YAAhB,OAAgB,gBAAhB,C;cAAGB,sC;cAAM,IuGt+BgE,UvGs+BiD,oBuGt+BkD,MAAK,EvGs+BrE,C;gBAAwB,qB;;YuGt+Bf,SAA4B,IvGu+BpD,KuGv+BoD,I;;UAA/C,yB;UzGqrCC,kB;UADb,YAAY,C;UACC,SyGprCK,aAAN,KAAM,CzGorCL,W;UAAb,OAAa,gBAAb,C;YAAa,wB;YyGnrCG,IzGmrCU,oBAAmB,cAAnB,EAAMb,sBAAnB,UyGnrCN,gBAAJ,C;cAA2B,aAAO,uB;YACIC,aAAO,gBzGkrCgC,IyGlrChC,CAAa,gBAAO,IAAP,C;;UAGxB,aAAO,gBAAO,KAAP,CAAC,gBAAO,IAAP,C;;QAGzB,aAAO,gBAAO,SAAP,CAAKB,gBAAO,IAAP,C;;MAG7B,iBAAiB,mC;MACjB,InIyHoD,CmIzHhD,UnIyHiD,UmIzHrD,C;QACI,uBAAuB,SAAS,M;QACtB,8B;QAAV,OAAU,gBAAV,C;UAAU,qB;UACJ,qBAAF,CAAE,EAAC,gBAAd,EAAGc,cAAhC,C;;MAGV,OAAO,I;K;yDAGX,6B;MAIwB,Q;MAHpB,mBAwB,C;MACxB,gBAAqB,C;MACrB,mBAAwB,C;MACJ,OrHyIjB,MAAO,KqHzIgb,eAAS,OAAT,GAakB,oBAAlB,IrHyIhB,EqHzIiD,KAAM,OAAN,GAAe,UAAf,IrHyIjD,C;MqHzIV,eAAY,CAAZ,oB;QACI,QAAQ,iBAAY,iBAAN,KAAM,CAAN,GAakB,GAAlB,IAAN,C;QACR,IAAI,MAAK,2BAakB,iBAAT,eAAS,CAAT,GAaqB,GAArB,IAAT,CAAT,C;UAA6C,K;QAC7C,IAAI,MAAK,EAAT,C;UACI,8BAAGB,CAAhB,I;UACA,eAAe,S;UACf,YAAY,G;;MAGpB,IAAI,gBAAgB,CAApB,C;QAAuB,OAAO,K;MAC9B,OAAO,eAAe,CAAF,IAAoB,iBAAY,iBAAN,KAAM,CAAN,IAAmB,YAAnB,GAakC,CAAlC,KAAN,MAA+C,EAAIE,C;QACI,8BAAGB,CAAhB,I;MAGJ,OAAa,YAAN,KAAM,EAAS,YAAT,CAAN,IAA+B,cAAW,eAAe,CAAF,IAAX,uCAA/B,C;K;;yHC/H+C,Y;MAAQ,W;K;IAEtE,gD;MACkB,UAMP,M;MANO,IAAI,aAAY,CAAhB,C;QACV,Y;;QAEA,UxBuZ8C,MAAW,KwBvZ/C,IxBuZ+C,EwBvZtC,QxBuZsC,C;QwBtZd,OAAA,IAAO,OxB2UmC,MAAW,KwB3UpC,KxB2UoC,CwB3UxC,GAAa,GAAnB,CAAP,GAAiC,GAAjC,GxBwV2C,WwBxVC,KxBwVD,C;;MwB5V/C,kB;MAMO,IxByUuC,MAAW,KwBzU1C,OxByU0C,CwBzU9C,GAAe,MAAnB,C;QAEmC,SAA9B,OAAAY,SAAQ,QAAR,C;;QAGpB,exBoU0C,MAAW,KwBpUIC,OxBoUkC,C;QwBnUrD,qBAA8B,QAAAY,axBgRC,MAAW,MAvCV,YwBzOqB,QxByOrB,CAuCU,CwBhRA,GAAwB,QAAPC,C;QAC1C,SAAI,UAAU,CAAd,GAAiB,MAAG,cAAPB,GAAYC,c;;M

AP7C,a;K;IAWJ,6C;MACI,OAAa,KAAY,gBA Ae,OAAf,EAAwB,MAAK,4BAA2B,QAA3B,CAAL,EAAxB,C;K;I  
CtBQ,4C;MAFrC,e;MAEsC,0B;MAFtC,iB;MAAA,uB;K;IAAA,mC;MAAA,sC;O;MAGI,uEAGY,GAHZ,C;MAIA  
,yEAGa,MAHb,C;MAIA,yEAGa,SAHb,C;MAIA,+DAGQ,KAHR,C;MAIA,+DAGQ,MAHR,C;MAIA,2DAGM,M  
AHN,C;MAIA,yDAGK,OAHL,C;K;IAxBA,gD;MAAA,yB;MAAA,wC;K;IAIA,iD;MAAA,yB;MAAA,yC;K;IAI  
A,iD;MAAA,yB;MAAA,yC;K;IAIA,4C;MAAA,yB;MAAA,oC;K;IAIA,4C;MAAA,yB;MAAA,oC;K;IAIA,0C;  
MAAA,yB;MAAA,kC;K;IAIA,yC;MAAA,yB;MAAA,iC;K;IA3BJ,+B;MAAA,4Q;K;IAAA,oC;MAAA,a;AAAA,  
a;UAAA,6C;aAAA,c;UAAA,8C;aAAA,c;UAAA,8C;aAAA,S;UAAA,yC;aAAA,S;UAAA,yC;aAAA,O;UAAA,uC;  
aAAA,M;UAAA,sC;UAAA,6D;K;IAiCA,4D;MAGW,Q;MADP,0BAA2C,iBAAjB,UAAW,cAAM,EAAU,UAA  
W,cAArB,C;MAEvC,0BAAsB,CAAtB,C;QAA2B,gBAAS,UAAW,cAAX,GAAMB,UAAW,cAAvC,C;WAC3B,0B  
AAsB,CAAtB,C;QAA2B,gBAAS,UAAW,cAAX,GAAMB,UAAW,cAAvC,C;QACnB,Y;MAHZ,W;K;IAOJ,oE;M  
AGW,Q;MADP,0BAA2C,iBAAjB,UAAW,cAAM,EAAU,UAAW,cAArB,C;MAEvC,0BAAsB,CAAtB,C;QAA2B,  
sBAA8C,uBAArC,UAAW,cAAX,GAAMB,UAAW,cAAO,CAA9C,C;WAC3B,0BAAsB,CAAtB,C;QAA2B,iBAA  
8C,uBAArC,UAAW,cAAX,GAAMB,UAAW,cAAO,CAA9C,C;QACnB,Y;MAHZ,W;K;IAOJ,8D;MAGW,Q;MA  
DP,0BAA2C,iBAAjB,UAAW,cAAM,EAAU,UAAW,cAArB,C;MAEvC,0BAAsB,CAAtB,C;QACI,YAAkD,uBAA  
rC,UAAW,cAAX,GAAMB,UAAW,cAAO,C;QACID,aAAa,eAAQ,KAAR,C;QAET,sBAAS,KAAT,GAaKB,KAAL  
B,E;UAA2B,a;AAC3B,uBAAQ,CAAR,C;::aAIR,0BAAsB,CAAtB,C;QAA2B,iBAA8C,uBAArC,UAAW,cAAX,G  
AAMB,UAAW,cAAO,CAA9C,C;QACnB,Y;MAXZ,W;K;::ICxCJ,+B;MAAA,mC;MAWiB,wB;MANT,aAAR,O  
AAO,OAAQ,KAAL,WAAY,IAAG,OAAO,SAAX,IAAwB,CAAC,CAAC,OAAO,SAAS,K;MADpE,sBAGQ,MAH  
R,GAIQ,iBAAa,OAAb,CAJR,GAMQ,qBACK,OADjB,OAAO,IAAK,KAAL,WAAJ,GAaKB,IAALB,GAAYB,UAA  
zB,4GAIO,+B;K;4CAGf,Y;MAA+C,OAAA,mBAAa,U;K;WDAC5D,oB;MAAqE,OAAA,mBAAa,qBAAY,QA AZ,  
C;K;0DACIF,8B;MACI,OAAA,mBAAa,uBAAc,QAAAd,EAAwB,QAAXB,C;K;IApBrB,2C;MAAA,0C;QAAA,yB  
;MAAA,mC;K;IA6B2B,+B;MAAC,wB;K;qCAExB,Y;MAAwC,8CAAc,cAAQ,SAAtB,C;K;iDACxC,oB;MAEmB  
,IAAS,I;MzHsDrB,QyHtDH,cAAQ,QAAO,eAAS,OAAT,QAAS,gBAAT,uBAAP,C;MACI,c3I/BT,EAAL,CAAJ,C;  
M2I+BkB,Y3IoElB,EAAL,CAAJ,C;M2ItEH,OAEuC,aAAR,OAAQ,qCAAR,aAAiD,aAAN,KAAM,yCAAjD,C;K;  
mDAEnC,8B;MAEK,IAAS,I;MzHiDP,QyHjDF,eAAS,OAAT,QAAS,gBAAT,uB;MAA0C,c3InCx,C,EAAL,CAAJ,  
C;M2ImCiD,Y3IgejD,EAAL,CAAJ,C;MUKlBW,uB;MAAP,eAAuB,6B;MiInpB9B,8CAGQ,CAAkB,YAAY,U1Bi  
QW,Y0BjQiB,6D1BiQjB,C0BjQvB,CAAhB,EAAOF,QAAQ,QAAIG,CAHR,C;K;sCAQJ,Y;MAAkC,qC;K;IAKF,  
4C;MAAC,8B;K;6CAEjC,Y;MAA6B,OAAA,gBAAY,M;K;8CAEzC,Y;MAAwC,8CAAc,aAd,C;K;0DACxC,oB;  
MAAwE,IAAS,I;MAAnB,OjI2CZ,aiI3Ca,iBAAS,QAAS,OAAT,QAAS,gBAAT,oCAAT,CjI2Cb,4B;K;4DiI1CID,8  
B;MAC8B,IAAS,I;MAAnC,8CAAc,YAAY,SAAS,OAAT,QAAS,gBAAT,wCAA6B,QAAS,0DAAID,CAAd,C;K;  
+CAEJ,Y;MAAkC,2C;K;IAGtC,6B;MAAA,iC;K;yCAGI,Y;MAA6B,OAAe,U;K;OCAE5C,Y;MAAwC,8CAAc,aA  
Ad,C;K;sDACxC,oB;MAAwE,IAAS,I;MAAnB,OjI8BZ,aiI9Ba,iBAAS,QAAS,OAAT,QAAS,gBAAT,oCAAT,CjI  
8Bb,4B;K;wDiI7BID,8B;MAC8B,IAAS,I;MAAnC,8CAAc,YAAY,SAAS,OAAT,QAAS,gBAAT,wCAA6B,QAAS  
,0DAAID,CAAd,C;K;2CAEJ,Y;MAAkC,+B;K;IAVtC,yC;MAAA,wC;QAAA,uB;MAAA,iC;K;IAaA,4B;MAA8  
D,IAAO,QAAPB,KAAoB,CAAP,C;QAAGB,MAAM,gCAAYB,uCAAzB,C;MAAnC,Y;K;ICzFjD,gD;MAQ+B,kB  
AAPB,wBAAc,IAAd,C;MAA0B,I1HgEjC,a;M0HhEA,O1HiEO,W;K;I0H9DX,gD;MAQqD,kBAA1B,gBAAhB,sC  
AAGB,EAAc,IAAd,EAAoB,IAAPB,C;MAAiC,sB1HoEID,W0HpEkD,C;MAAxD,O1HqEO,W;K;I2HzFX,yC;MA  
EkD,8B;MAAA,OCGN,aDHwB,yBAAa,QAAb,mCCGxB,C5G+xBgC,sB;K;I2GhyB5E,2C;M7IugIW,kBAAY,gB;  
MAoGH,Q;MAAhB,wB6IpmIqB,U7IomIrB,gB;QAAGB,c6IpmIk,U7IomIrB,M;QAAsB,IAAI,C6IpmIkB,sB7IomI  
P,O6IpmIO,C7IomItB,C;UAAYB,WAAY,WAAI,OAAJ,C;M6IpmI3D,qB7IqmIO,W;M6IpmIP,IxIkNwD,CwIlNp  
D,cxIkNqD,UwIlNzD,C;Q3GgKuC,U;Q2G/JnC,qB3G+JyD,OAAtB,+B2G/Jd,mB3G+Jc,uBAAsB,CAAOW;QmG  
kO7C,kBAAhB,sB;QQ/XC,0C;QACA,IAAI,E3G8QoC,0BAAU,C2G9Q9C,CAAJ,C;UACI,2BAAO,GAAP,C;QA  
EW,sCAAa,GAAb,C;QALnB,sB3H4DG,WmHoUqC,W;QQzXxC,OAAO,I;MAGX,OAAO,K;K;IAGX,8C;MAO  
mB,c;Q7Iw3YC,Q;QAAbB,wB6Ix3YI,U7Iw3YJ,gB;UAAgB,c6Ix3YZ,U7Iw3YJ,M;UAAsB,I6Ix3YD,sB7Iw3Ye,  
O6Ix3Yf,C7Iw3YC,C;YAAwB,aAAO,I;YAAP,e;QAC9C,aAAO,K;M6Iz3YP,e;QACI,kBAA6B,MAAX,UAAW  
,C;Q3GyIM,U;Q2GxIb,a3GwImC,OAAtB,+B2GxIvB,mB3GwIuB,uBAAsB,CAAOW;Q2GxIX,kBC/BjB,aD+BD,  
MC/BC,C5Gg1C6C,uBAAzB,CAAYB,C;QbjmB9E,kBAAS,gB;QA2FA,U;QAAA,+B;QAAbB,OAAgB,gBAAhB,  
C;UAAgB,6B;UAAM,IwH3yB4C,4BxH2yB9B,SwH3yB8B,CxH2yB5C,C;YAAwB,WAAY,WAAI,SAAJ,C;QwH



3yBtD,sBAAmF,exH4yBhF,WwH5yBgF,EAAa,GAAb,C;QACnF,OAAO,I;;MAGX,OAAO,K;K;IEncP,iC;MAAQ,8BAAY,IAAK,UAAjB,IAA8B,uBAAY,IAAK,mB;K;IAOvD,oC;MAAQ,8BAAY,IAAK,a;K;ICZ7B,4B;MAGI,OAAO,yBAAP,C;QACI,sBAAY,mCAAZ,C;;K;IAIR,uC;MAOI,sBAAY,sCAAgB,gBA Ae,IAAf,CAA5B,C;MACA,OAAO,S;K;ICbP,4B;MAAQ,mB;K;IACR,mC;MACI,eAAO,K;K;IAKX,4B;MAAQ,mB;K;IACR,mC;MACI,eAAO,K;K;iHCoBf,sJ;MAEyC,qB;QAAA,QAakB,I;MAAM,qB;QAAA,QAakB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,yB;QAAA,YAAaB,I;MAAM,kC;QAAA,qBAA+B,I;MAAM,qC;QAAA,wBAaK,C,K;MAAO,+C;QAAA,kCAA4C,K;MAAO,4C;QAAA,+BAAYC,K;MACtT,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,WAAF,IAAiB,S;MACjB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,uBAAF,IAA6B,qB;MAC7B,EAAE,iCAAF,IAAuC,+B;MACvC,EAAE,8BAAF,IAAoC,4B;MACpC,OAAO,C;K;+GAw0BX,wD;MAEwC,6B;QAAA,gBAAYB,E;MAAI,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC/I,QAAQ,E;MACR,EAAE,eAAF,IAAqB,a;MACrB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;6EA6CX,4B;MAE6D,iBAAY,KA AZ,C;K;6EAE7D,mC;MAEoE,UAA Y,KA AZ,IAAqB,K;K;6EAuBzF,4B;MAE8D,iBAAY,KA AZ,C;K;6EAE9D,mC;MAEqE,UAA Y,KA AZ,IAAqB,K;K;6EAuB1F,4B;MAEqE,iBAAY,KA AZ,C;K;6EAErE,mC;MAE4E,UAA Y,KA AZ,IAAqB,K;K;6EAuBjG,4B;MAE+D,iBAAY,KA AZ,C;K;6EAE/D,mC;MAEsE,UAA Y,KA AZ,IAAqB,K;K;6EAuB3F,4B;MAEgE,iBAAY,KA AZ,C;K;6EAEhE,mC;MAEuE,UAA Y,KA AZ,IAAqB,K;K;6EAuB5F,4B;MAE6D,iBAAY,KA AZ,C;K;6EAE7D,mC;MAEoE,UAA Y,KA AZ,IAAqB,K;K;6EAuBzF,4B;MAE8D,iBAAY,KA AZ,C;K;6EAE9D,mC;MAEqE,UAA Y,KA AZ,IAAqB,K;K;6EAuB1F,4B;MAEiE,iBAAY,KA AZ,C;K;6EAEjE,mC;MAEwE,UAA Y,KA AZ,IAAqB,K;K;8EAuB7F,4B;MAEkE,iBAAY,KA AZ,C;K;6EAEIE,mC;MAEyE,UAA Y,KA AZ,IAAqB,K;K;6GC3oC9F,wD;MAEqC,6B;QAAA,gBAA+B,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACpJ,QAAQ,E;MACR,EAAE,eAAF,IAAqB,a;MACrB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;mIAiCX,+B;MAEgD,mC;QAAA,sBAAGC,K;MAC5E,QAAQ,E;MACR,EAAE,qBAAF,IAA2B,mB;MAC3B,OAAO,C;K;4EC9CX,4B;MAEgE,iBAAY,KA AZ,C;K;4EAgChE,4B;MAEyE,iBAAY,KA AZ,C;K;4EAIbZ E,4B;MAEmE,iBAAY,KA AZ,C;K;4EAYnE,4B;MAEOE,iBAAY,KA AZ,C;K;oIC7a1E,4H;MAE8C,qB;QAAA,QAAiB,E;MAAI,6B;QAAA,gBAAgC,E;MAAW,iC;QAAA,oBAA2D,E;MAAW,iC;QAAA,oBAA2D,E;MAAW,qC;QAAA,wBAmJvJ,U;;MANJqO,+B;QAAA,kBA mJrO,U;;MANJ6S,4B;QAAA,eAA+B,S;MAC3a,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,eAAF,IAAqB,a;MACrB,EAAE,mBAAF,IAAyB,iB;MACzB,EAAE,mBAAF,IAAyB,iB;MACzB,EAAE,uBAAF,IAA6B,qB;MAC7B,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,cAAF,IAAoB,Y;MACpB,OAAO,C;K;wIAYX,mC;MAEgD,2B;QAAA,cAAuB,E;MAAI,0B;QAAA,aAAsB,E;MAC7F,QAAQ,E;MACR,EAAE,aAAF,IAAmB,W;MACnB,EAAE,YAAF,IAAkB,U;MACIB,OAAO,C;K;8HakEX,+D;MAEqG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC/K,QAAQ,E;MACR,EAAE,aAAF,IAAmB,W;MACnB,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;4HAWBX,iE;MAE0C,4B;QAAA,eAAwB,E;MAAI,wB;QAAA,WAAyB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC/K,QAAQ,E;MACR,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;sGAUqE,qB;MAAQ,OAAW,U;K;sGAEnB,qB;MAAQ,OAAW,U;K;4GAehB,qB;MAAQ,OAAc,a;K;wGAS1B,qB;MAAQ,OAA Y,W;K;0HAEX,qB;MAAQ,OAAqB,oB;K;kGASnD,qB;MAAQ,OAA S,Q;K;oGAehB,qB;MAAQ,OAAU,S;K;sGAEjB,qB;MAAQ,OAAW,U;K;wHAEV,qB;MAAQ,OAAoB,mB;K;wHAE5B,qB;MAAQ,OAAoB,mB;K;kHAE/B,qB;MAAQ,OAAiB,gB;K;kHAEzB,qB;MAAQ,OAAiB,gB;K;oHASd,qB;MAAQ,OAAkB,iB;K;oHAE1B,qB;MAAQ,OAAkB,iB;K;oHAE1B,qB;MAAQ,OAAkB,iB;K;wIAEhB,qB;MAAQ,OAA4B,2B;K;4FC1MnI,uD;MAE8B,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACHJ,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;kGAuBX,sE;MAEiC,6B;QAAA,gBAA8B,I;MAAM,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACvL,QAAQ,E;MACR,EAAE,eAAF,IAAqB,a;MACrB,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;M

AChB,OAAO,C;K;kGA8DX,8U;MAEiC,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,sB;QAAA,SAAiB,C;MAAG,uB;QAAA,UAAkB,C;MAAG,6B;QAAA,gBAA8B,I;MAAM,sB;QAAA,SAAkB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC3wB,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,eAAF,IAAqB,a;MACrB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;wGAgDX,kQ;MAEoC,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC7IB,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;kGAsCX,iX;MAEiC,sB;QAAA,SA AkB,G;MAAK,sB;QAAA,SAAkB,G;MAAK,sB;QAAA,SAAkB,G;MAAK,yB;QAAA,YAAkB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,sB;QAAA,SAAiB,C;MAAG,uB;QAAA,UAAkB,C;MAAG,6B;QAAA,gBAA8B,I;MAAM,sB;QAAA,SAAkB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACr2B,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,QAAF,IAAc,M;MACd,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,eAAF,IAAqB,a;MACrB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;kGA2BX,0E;MAEiC,oB;QAAA,OAAgB,E;MAAI,2B;QAAA,cAAwB,K;MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACtM,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,aAAF,IAAmB,W;MACnB,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;M

ACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;wGAmDX,4S;MAEoC,mB;QAAA,MAAe,E;MAAI,oB;QAA  
A,OAAgB,E;MAAI,wB;QAAA,WAAiB,C;MAAG,sB;QAAA,SAAmB,K;MAAO,2B;QAAA,cAAwB,K;MAAO,u  
B;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;  
MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,i  
BAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,  
6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAgB,  
I;MAAM,sB;QAAA,SAAe,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WA  
AqB,K;MACjtB,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE,MAAF,IAAY,I;MACZ,EAAE,UAAF,IA  
AgB,Q;MACHB,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,SAAF,IAAe,O;MACf,EA  
AE,UAAF,IAAgB,Q;MACHB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB  
,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,gBAAF,IAAsB,c;MACt  
B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,eA  
AF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAAY,I;  
MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAA  
F,IAAgB,Q;MACHB,OAAO,C;K;8GAuBX,6D;MAEuC,oB;QAAA,OAAgB,E;MAAI,oB;QAAA,OAAgB,I;MAA  
M,sB;QAAA,SAAe,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K  
;MAC7K,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;M  
ACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;  
wECnbX,4B;MAEyE,iBAAY,KAAZ,C;K;wEAEzE,2B;MAEgG,iBAAY,IAAZ,C;K;wEAwBhG,oC;MAE+F,UAA  
Y,KAAZ,IAAqB,M;K;wEAmFpH,2B;MAEqE,iBAAY,IAAZ,C;K;wEAErE,kC;MAE2E,UAA,Y,IAAZ,IAAoB,K;K  
;wEAssC/F,4B;MAEyE,iBAAY,KAAZ,C;K;wEA0BzE,4B;MAEyE,iBAAY,KAAZ,C;K;wEAsBzE,4B;MAEuE,iB  
AAY,KAAZ,C;K;wEAyBvE,4B;MAE6E,iBAAY,KAAZ,C;K;2FA4C7E,gD;MAEiC,qB;QAAA,QAAiD,I;MAAM,  
uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACiK,QAAQ,E;MACR,EA  
AE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;  
MACHB,OAAO,C;K;uEA+UX,4B;MAEuE,iBAAY,KAAZ,C;K;wEAevE,2B;MAE6F,iBAAY,IAAZ,C;K;wEAqN  
7F,4B;MAEyE,iBAAY,KAAZ,C;K;wEAEzE,oC;MAE2F,UAA,Y,KAAZ,IAAqB,M;K;+FAuehH,wD;MAEmC,6B;  
QAAA,gBAA8B,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;M  
ACjJ,QAAQ,E;MACR,EAAE,eAAF,IAAqB,a;MACrB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MA  
CIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;uGAuIX,mB;MAEuC,uB;QAAA,UAAoB,K;MACvD,QAAQ,E;  
MACR,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;+HAyCX,iB;MAEmD,qB;QAAA,QAAkB,I;MACjE,QAAQ,E;M  
ACR,EAAE,OAAF,IAAa,K;MACb,OAAO,C;K;+FA0MX,sE;MAEmC,oB;QAAA,OAAgB,I;MAAM,wB;QAAA,  
WA0+G4B,S;MA1+GwB,kB;QAAA,KAAc,E;MAAI,wB;QAAA,WAAoB,I;MAAM,sB;QAAA,SAAkB,S;MAA  
W,uB;QAAA,UAAoB,I;MAAM,qB;QAAA,QAAiB,I;MAAM,oB;QAAA,OAAgB,I;MACnP,QAAQ,E;MACR,EA  
AE,MAAF,IAAY,I;MACZ,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,IAAF,IAAU,E;MACV,EAAE,UAAF,IAAgB,  
Q;MACHB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,OAAF,IAAa,K;MACb,EAAE,MA  
AF,IAAY,I;MACZ,OAAO,C;K;qIAgDX,iB;MAEsD,qB;QAAA,QAAkB,I;MACpE,QAAQ,E;MACR,EAAE,OAA  
F,IAAa,K;MACb,OAAO,C;K;+GAKBX,qB;MAE2C,yB;QAAA,YAAmB,S;MAC1D,QAAQ,E;MACR,EAAE,SA  
F,IAAe,S;MACf,OAAO,C;K;wEAkCX,4B;MAEqF,iBAAY,KAAZ,C;K;yFAgCrF,4V;MAEgC,4B;QAAA,eAA8B,  
I;MAAM,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,U  
AAgB,C;MAAG,sB;QAAA,SAAiB,C;MAAG,uB;QAAA,UAAkB,C;MAAG,6B;QAAA,gBAA8B,I;MAAM,sB;Q  
AAA,SAAkB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MA  
AO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aA  
AuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;  
QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K  
;MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SAAe,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAu  
B,K;MAAO,wB;QAAA,WAAqB,K;MAC9yB,QAAQ,E;MACR,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,SAAF,IA  
Ae,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,QAA  
F,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,eAAF,IAAqB,a;MACrB,EAAE,QAAF,IAAc,M;MACd,EA

AE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;  
MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACIB,  
EAAE,gBAAF,IAAsB,c;MACTb,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,  
IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACTb,EAAE,oBAAF,IAA0B,kB;  
MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,  
IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;wEAwEX,2B;MAE+D,iBAAY,IAAZ,C;K;iGA2  
D/D,gD;MAEoC,qB;QAAA,QAAC,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QA  
AA,WAAqB,K;MACII,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAA  
F,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;qGA2BX,yD;MAEsC,sB;QAAA,SAAkB,E;M  
AAI,sB;QAAA,SAAkB,E;MAAI,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB  
,K;MAC5J,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;  
MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;6GAuBX,oD;MAE0C,yB;Q  
AAA,YAAsB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC  
jJ,QAAQ,E;MACR,EAAE,WAAF,IAAiB,S;MACjB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACI  
B,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;2FAoFX,kF;MAEiC,uB;QAAA,UAAmB,E;MAAI,wB;QAAA,W  
AAoB,E;MAAI,sB;QAAA,SAAc,C;MAAG,qB;QAAA,QAAC,C;MAAG,qB;QAAA,QAAC,I;MAAM,uB;QAAA,U  
AAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACjN,QAAQ,E;MACR,EAAE,SAAF,IA  
Ae,O;MACf,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,QAAF,IAAc,M;MACd,EAAE,OAAF,IAAa,K;MACb,EAAE  
,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;M  
ACHb,OAAO,C;K;iHAYBX,0D;MAEqE,sB;QAAA,SAAc,S;MAAW,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aA  
AuB,K;MAAO,wB;QAAA,WAAqB,K;MACzK,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAA  
c,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAA  
O,C;K;wEAmXX,4B;MAEkE,iBAAY,KAAZ,C;K;wEAEIE,2B;MAEoE,iBAAY,IAAZ,C;K;wEAUpE,4B;MAEsE,  
iBAAY,KAAZ,C;K;wEAEtE,2B;MAEwE,iBAAY,IAAZ,C;K;wEAaxE,4B;MAE+D,iBAAY,KAAZ,C;K;wEAE/D,  
2B;MAEiE,iBAAY,IAAZ,C;K;mGA0CjE,8G;MAEqC,gC;QAAA,mBAooF8C,M;MApoFe,gC;QAAA,mBAmpFT  
,S;MANpFyE,oC;QAAA,uBA8pFjE,S;MA9pF6I,2B;QAAA,cAAoB,S;MAAW,4B;QAAA,eAAqB,S;MAAW,6B;  
QAAA,gBAYqFIO,K;MAxqFvE,QAAQ,E;MACR,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB  
;MACxB,EAAE,sBAAF,IAA4B,oB;MAC5B,EAAE,aAAF,IAAmB,W;MACnB,EAAE,cAAF,IAAoB,Y;MACpB,E  
AAE,eAAF,IAAqB,a;MACrB,OAAO,C;K;+FAwCX,mF;MAEmC,oB;QAAA,OAAa,I;MAAM,sB;QAAA,SAAkB,  
E;MAAI,2B;QAAA,cAAuB,E;MAAI,sB;QAAA,SAAY,C,I;MAAM,qB;QAAA,QAA6B,E;MAAW,uB;QAAA,UA  
AoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACxQ,QAAQ,E;MACR,EAAE,MAAF,IA  
AY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,QAAF,IAAc,M;MACd,EAA  
E,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;  
MACHb,OAAO,C;K;6FA4BX,2B;MAEkC,+B;QAAA,kBAA4B,K;MAC1D,QAAQ,E;MACR,EAAE,iBAAF,IAA  
uB,e;MACvB,OAAO,C;K;2FA2DX,iE;MAEiC,wB;QAAA,WAAqB,K;MAAO,oB;QAAA,OAAe,C;MAAG,sB;Q  
AAA,SAAkB,E;MAAI,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC/  
K,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MAC  
d,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;yF  
A8FX,6B;MAEgC,oB;QAAA,OA+7E6C,S;MA/7EL,2B;QAAA,cC12He,M;MDm2HnF,QAAQ,E;MACR,EAAE,  
MAAF,IAAY,I;MACZ,EAAE,aAAF,IAAmB,W;MACnB,OAAO,C;K;wEAoDX,0B;MAE+D,iBAAY,GAZ,C;K;  
wEAE/D,iC;MAEqE,UAY,GAZ,IAAmB,K;K;+FAoDxF,oF;MAEmC,mB;QAAA,MAAe,I;MAAM,wB;QAAA,  
WAAoB,I;MAAM,wB;QAAA,WAAoB,I;MAAM,mB;QAAA,MAAe,E;MAAI,2B;QAAA,cAAwB,I;MAAM,uB;Q  
AAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACvO,QAAQ,E;MACR,EAAE,K  
AAF,IAAW,G;MACX,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,KAAF,IAAW,G  
;MACX,EAAE,aAAF,IAAmB,W;MACnB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,U  
AAF,IAAgB,Q;MACHb,OAAO,C;K;iFAwNX,yC;MAE4B,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;M  
AAO,wB;QAAA,WAAqB,K;MACtG,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MA  
CIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;6FAwBX,iD;MAEkC,sB;QAAA,SAAc,I;MAAM,uB;QAAA,U

AAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACjI,QAAQ,E;MACR,EAAE,QAAF,IA  
Ac,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OA  
AO,C;K;uGASX,mB;MAEuC,uB;QAAA,UAAoB,K;MACvD,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,OA  
AO,C;K;6GAYX,kC;MAE0C,uB;QAAA,UAAoB,K;MAAO,oB;QAAA,OAAiB,K;MAAO,uB;QAAA,UAAoB,K;  
MAC7G,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,MAAF,IAAY,I;MACZ,EAAE,SAAF,IAAe,O;MA  
Cf,OAAO,C;K;wEAkEX,4B;MAE6D,iBAAY,KAAZ,C;K;wEAU7D,4B;MAEsE,iBAAY,KAAZ,C;K;wEAEtE,2B;  
MAEwE,iBAAY,IAAZ,C;K;uGAsCxE,oH;MAEuC,yB;QAAA,YAAsB,K;MAAO,0B;QAAA,aAAuB,S;MAAW,6  
B;QAAA,gBAA0B,S;MAAW,uB;QAAA,UAAoB,K;MAAO,iC;QAAA,oBAA8B,S;MAAW,qC;QAAA,wBAAkC,  
S;MAAW,+B;QAAA,kBAAkC,S;MAC1R,QAAQ,E;MACR,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAk  
B,U;MACIB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,SAAF,IAAe,O;MACf,EAAE,mBAAF,IAAyB,iB;MACzB,EA  
AE,uBAAF,IAA6B,qB;MAC7B,EAAE,iBAAF,IAAuB,e;MACvB,OAAO,C;K;mGAgFX,oB;MAEqC,wB;QAAA,  
WAAqB,K;MACtD,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;wEA+MX,2B;MAEiE,iBAA  
Y,IAAZ,C;K;2GAKcJE,c;MAEyC,kB;QAAA,KAAgB,S;MACrD,QAAQ,E;MACR,EAAE,IAAF,IAAU,E;MACV,  
OAAO,C;K;2FAuMX,gB;MAGI,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;wEAgBX,4B;MAEi  
E,iBAAY,KAAZ,C;K;wEAEjE,oC;MAE4E,iBAAY,aAAZ,C;K;wEAuT5E,4B;MAEmE,iBAAY,KAAZ,C;K;uFA2  
CnE,sB;MAE+B,iB;QAAA,IAAa,G;MAAK,iB;QAAA,IAAa,G;MAAK,iB;QAAA,IAAa,G;MAAK,iB;QAAA,IAA  
a,G;MAC9F,QAAQ,E;MACR,EAAE,GAAF,IAAS,C;MACT,EAAE,GAAF,IAAS,C;MACT,EAAE,GAAF,IAAS,C  
;MACT,EAAE,GAAF,IAAS,C;MACT,OAAO,C;K;qFA0CX,+B;MAE8B,iB;QAAA,IAAa,G;MAAK,iB;QAAA,IA  
Aa,G;MAAK,qB;QAAA,QAAiB,G;MAAK,sB;QAAA,SAAkB,G;MACtG,QAAQ,E;MACR,EAAE,GAAF,IAAS,C  
;MACT,EAAE,GAAF,IAAS,C;MACT,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,OAAO,C;K;w  
EAOX,4B;MAEmE,iBAAY,KAAZ,C;K;yFAiHnE,oB;MAEgC,wB;QAAA,WAY2B+C,M;;MAx2B3E,QAAQ,E;M  
ACR,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;6FAeX,+B;MAEKc,oB;QAAA,OAAgB,S;MAAW,mB;QAAA  
,MAAe,S;MAAW,wB;QAAA,WAq1BR,M;;MAP1B3E,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,K  
AAF,IAAW,G;MACX,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;6GAwCX,yD;MAE0C,qB;QAAA,QAAiB,E;  
MAAI,uB;QAAA,UAAoB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WA  
AqB,K;MACpK,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,  
O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;yGAiCX,mC;MAEwC,q  
B;QAAA,QA2wByD,Q;;MA3wBK,sB;QAAA,SA2wBL,Q;;MA3wBoE,wB;QAAA,WA4vBtF,M;;MA3vB3E,QAA  
Q,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,UAAF,IAAgB,Q;MACHb,OAAO  
,C;K;2FAYX,2B;MAEiC,mB;QAAA,MAuwB0C,Q;;MAvwBJ,0B;QAAA,aAAsB,S;MACzF,QAAQ,E;MACR,EA  
AE,KAAF,IAAW,G;MACX,EAAE,YAAF,IAAkB,U;MACIB,OAAO,C;K;+GAYX,0B;MAE2C,uB;QAAA,UAAq  
BgC,Q;;MARvBU,qB;QAAA,QAqvBV,Q;;MAPvBvE,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,OAA  
F,IAAa,K;MACb,OAAO,C;K;wEAgCX,4B;MAE+D,iBAAY,KAAZ,C;K;qFAyaY,qB;MAAQ,OAAU,S;K;6FAEd,  
qB;MAAQ,OAAc,a;K;uFAEzB,qB;MAAQ,OAAW,U;K;iFASxB,qB;MAAQ,OAAQ,E;K;iFAEX,qB;MAAQ,OAA  
Q,O;K;uFAEb,qB;MAAQ,OAAW,U;K;uFAS3B,qB;MAAQ,OAAW,U;K;mFAErB,qB;MAAQ,OAAS,Q;K;qFAEh  
B,qB;MAAQ,OAAU,S;K;yFAShB,qB;MAAQ,OAAW,W;K;uFAErB,qB;MAAQ,OAAW,U;K;+FAEf,qB;MAAQ,O  
AAe,c;K;uFAE3B,qB;MAAQ,OAAW,U;K;uFAEnB,qB;MAAQ,OAAW,U;K;mFASrB,qB;MAAQ,OAAS,Q;K;iF  
AEiB,qB;MAAQ,OAAQ,O;K;6EAEiB,qB;MAAQ,OAAM,K;K;uFAET,qB;MAAQ,OAAW,U;K;qFASiB,qB;MAA  
Q,OAAU,S;K;qFAEiB,qB;MAAQ,OAAU,S;K;6EASR,qB;MAAQ,OAAM,K;K;mFAEX,qB;MAAQ,OAAS,Q;K;+  
EAEnB,qB;MAAQ,OAAO,M;K;+EAS/B,qB;MAAQ,OAAO,M;K;iFAEd,qB;MAAQ,OAAQ,O;K;mFAEf,qB;MA  
AQ,OAAS,Q;K;mFAShB,qB;MAAQ,OAAQ,O;K;iFAEhB,qB;MAAQ,OAAQ,O;K;iFAEhB,qB;MAAQ,OAAQ,O;  
K;mFASd,qB;MAAQ,OAAQ,O;K;+EAEiB,qB;MAAQ,OAAM,K;K;+EAEb,qB;MAAQ,OAAO,M;K;iFAEd,qB;M  
AAQ,OAAQ,O;K;mFAEf,qB;MAAQ,OAAS,Q;K;6EASd,qB;MAAQ,OAAM,K;K;qFAEV,qB;MAAQ,OAAU,S;K  
;mFAEnB,qB;MAAQ,OAAS,Q;K;2FAEb,qB;MAAQ,OAAa,Y;K;6FAEpB,qB;MAAQ,OAAc,a;K;mFAE3B,qB;M  
AAQ,OAAS,Q;K;6EAS1B,qB;MAAQ,OAAM,K;K;6EAEd,qB;MAAQ,OAAM,K;K;qFAEV,qB;MAAQ,OAAU,S;  
K;+EASjB,qB;MAAQ,OAAO,M;K;mFAEb,qB;MAAQ,OAAS,Q;K;+EASrB,qB;MAAQ,OAAO,M;K;iFAEd,qB;  
MAAQ,OAAQ,O;K;iFASjB,qB;MAAQ,OAAO,M;K;6FAER,qB;MAAQ,OAAc,a;K;qFAE1B,qB;MAAQ,OAAU,S  
;K;iFASb,qB;MAAQ,OAAO,M;K;uFAEZ,qB;MAAQ,OAAU,S;K;yFAS9B,qB;MAAQ,OAAW,W;K;+EAE1B,qB;

MAAQ,OAAM,K;K;qFAEX,qB;MAAQ,OAAS,Q;K;iFAEnB,qB;MAAQ,OAAO,M;K;+EASrB,qB;MAAQ,OAAO ,M;K;6FAER,qB;MAAQ,OAAC,a;K;qFAS1B,qB;MAAQ,OAAU,S;K;mFAEnB,qB;MAAQ,OAAS,Q;K;+EASX,q B;MAAQ,OAAO,M;K;mFAEb,qB;MAAQ,OAAS,Q;K;iFASnB,qB;MAAQ,OAAO,M;K;qFAEZ,qB;MAAQ,OAA U,S;K;mFAEnB,qB;MAAQ,OAAS,Q;K;kFASJ,qB;MAAQ,OAAQ,O;K;oFAEf,qB;MAAQ,OAAS,Q;K;8EAEpB,q B;MAAQ,OAAM,K;K;oFAEV,qB;MAAQ,OAAU,S;K;mFASzC,qB;MAAQ,OAAS,Q;K;mFAEjB,qB;MAAQ,OA AS,Q;K;qFAEhB,qB;MAAQ,OAAU,S;K;qFAElB,qB;MAAQ,OAAU,S;K;wIEx+M7E,wM;MAEiD,qB;QAAA,QA AkB,I;MAAM,sB;QAAA,SAAmB,I;MAAM,2B;QAAA,cAAwB,I;MAAM,yB;QAAA,YAAsB,I;MAAM,0B;QAA A,aAAuB,I;MAAM,0B;QAAA,aAAuB,I;MAAM,sB;QAAA,SAAmB,I;MAAM,0B;QAAA,aAAuB,I;MAAM,0B;Q AAA,aAAuB,I;MAAM,gC;QAAA,mBAA6B,I;MAAM,+B;QAAA,kBAA4B,I;MAAM,gC;QAAA,mBAA6B,I;MA AM,uB;QAAA,UAAoB,I;MAAM,4B;QAAA,eAAyB,I;MAAM,wB;QAAA,WAAqB,I;MAAM,uB;QAAA,UAAoB ,I;MACrf,QAAQ,E;MACr,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W; MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MACiB,EAAE,YAAF,IAAkB,U;MACiB,EAA E,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MACiB,EAAE,YAAF,IAAkB,U;MACiB,EAAE,kBAAF,IAAw B,gB;MACxB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,SAAF,IAAe,O;MACf, EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;wHA sDX,wM;MAEyC,qB;QAAA,QAAqB,S;MAAW,sB;QAAA,SAAsB,S;MAAW,2B;QAAA,cAA4B,S;MAAW,yB;Q AAA,YAA0B,S;MAAW,0B;QAAA,aAA6B,S;MAAW,0B;QAAA,aAA6B,S;MAAW,sB;QAAA,SAAuB,S;MAA W,0B;QAAA,aAA0B,S;MAAW,0B;QAAA,aAA0B,S;MAAW,gC;QAAA,mBAAoC,S;MAAW,+B;QAAA,kBAA mC,S;MAAW,gC;QAAA,mBAAoC,S;MAAW,uB;QAAA,UAAwB,S;MAAW,4B;QAAA,eAA4B,S;MAAW,wB;Q AAA,WAAoB,S;MAAW,uB;QAAA,UAAmB,S;MACtnB,QAAQ,E;MACr,EAAE,OAAF,IAAa,K;MACb,EAAE, QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB, U;MACiB,EAAE,YAAF,IAAkB,U;MACiB,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MACiB,EAA E,YAAF,IAAkB,U;MACiB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kBAAF,I AAwB,gB;MACxB,EAAE,SAAF,IAAe,O;MACf,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MACH B,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;SHAYX,kN;MAEWc,wB;QAAA,WAA4C,S;MAAW,qB;QAAA,QAA iB,S;MAAW,sB;QAAA,SAAkB,S;MAAW,2B;QAAA,cAAuB,S;MAAW,yB;QAAA,YAAqB,S;MAAW,0B;QAA A,aAsB,S;MAAW,0B;QAAA,aAsB,S;MAAW,sB;QAAA,SAAkB,S;MAAW,0B;QAAA,aAsB,S;MAAW,0B; QAAA,aAsB,S;MAAW,gC;QAAA,mBAA4B,S;MAAW,+B;QAAA,kBAA2B,S;MAAW,gC;QAAA,mBAA4B,S; MAAW,uB;QAAA,UAAmB,S;MAAW,4B;QAAA,eAAwB,S;MAAW,wB;QAAA,WAAoB,S;MAAW,uB;QAAA, UAAmB,S;MAC9IB,QAAQ,E;MACr,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,OAAF,IAAa,K;MACb,EAAE,QA AAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;M ACiB,EAAE,YAAF,IAAkB,U;MACiB,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MACiB,EAAE,YA AF,IAAkB,U;MACiB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kBAAF,IAAw B,gB;MACxB,EAAE,SAAF,IAAe,O;MACf,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MACHB,EA AE,SAAF,IAAe,O;MACf,OAAO,C;K;0HAsDX,wM;MAE0C,qB;QAAA,QAAiB,S;MAAW,sB;QAAA,SAAkB,S; MAAW,2B;QAAA,cAAuB,S;MAAW,yB;QAAA,YAAqB,S;MAAW,0B;QAAA,aAsB,S;MAAW,0B;QAAA,aA AsB,S;MAAW,sB;QAAA,SAAkB,S;MAAW,0B;QAAA,aAsB,S;MAAW,0B;QAAA,aAsB,S;MAAW,gC;QAA A,mBAA4B,S;MAAW,+B;QAAA,kBAA2B,S;MAAW,gC;QAAA,mBAA4B,S;MAAW,uB;QAAA,UAAmB,S;M AAW,4B;QAAA,eAAwB,S;MAAW,wB;QAAA,WAAoB,S;MAAW,uB;QAAA,UAAmB,S;MACziB,QAAQ,E;M ACR,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAA F,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MACiB,EAAE,YAAF,IAAkB,U;MACiB,EAAE,QAAF,IAAc,M;MA Cd,EAAE,YAAF,IAAkB,U;MACiB,EAAE,YAAF,IAAkB,U;MACiB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,i BA AF,IAAuB,e;MACvB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,SAAF,IAAe,O;MACf,EAAE,cAAF,IAAoB, Y;MACpB,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;gHAYDX,wM;MAEqC,qB; QAAA,QAAc,S;MAAW,sB;QAAA,SA Ae,S;MAAW,2B;QAAA,cAAuB,S;MAAW,yB;QAAA,YAAqB,S;MAAW, 0B;QAAA,aAsB,S;MAAW,0B;QAAA,aAsB,S;MAAW,sB;QAAA,SAAkB,S;MAAW,0B;QAAA,aAAmB,S;M AAW,0B;QAAA,aAAmB,S;MAAW,gC;QAAA,mBAA6B,S;MAAW,+B;QAAA,kBAA4B,S;MAAW,gC;QAAA, mBAA6B,S;MAAW,uB;QAAA,UAAmB,S;MAAW,4B;QAAA,eAAqB,S;MAAW,wB;QAAA,WAAoB,S;MAAW,

uB;QAAA,UAAmB,S;MACxhB,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MACiB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MACIB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,SAAF,IAAe,O;MACf,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;8HAqBX,gD;MAEsE,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACHJ,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;sIAoBX,gD;MAEgD,qB;QAAA,QAAiB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACjJ,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;wHAWCX,wB;MAEyC,qB;QAAA,QAAiB,K;MAAO,qB;QAAA,QAAiB,K;MAC9E,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,OAAO,C;K;kGAyBX,oB;MAE8B,mB;QAAA,MAAe,S;MAAW,mB;QAAA,MAAe,S;MACnE,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE,KAAF,IAAW,G;MACX,OAAO,C;K;oHAYX,kC;MAEuC,qB;QAAA,QAAiB,S;MAAW,qB;QAAA,QAAiB,S;MAAW,mB;QAAA,MAAe,S;MAAW,mB;QAAA,MAAe,S;MACpI,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,KAAF,IAAW,G;MACX,EAAE,KAAF,IAAW,G;MACX,OAAO,C;K;gGAYX,oB;MAE6B,mB;QAAA,MAAY,S;MAAW,mB;QAAA,MAAY,S;MAC5D,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE,KAAF,IAAW,G;MACX,OAAO,C;K;kHAYX,kC;MAEsC,qB;QAAA,QAAc,S;MAAW,qB;QAAA,QAAc,S;MAAW,mB;QAAA,MAAY,S;MAAW,mB;QAAA,MAAY,S;MACvH,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,KAAF,IAAW,G;MACX,EAAE,KAAF,IAAW,G;MACX,OAAO,C;K;gIAeX,wB;MAE6C,qB;QAAA,QAAkB,S;MAAW,qB;QAAA,QAAkB,S;MACxF,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,OAAO,C;K;oIAeX,wB;MAE+C,qB;QAAA,QAAiB,S;MAAW,qB;QAAA,QAAiB,S;MACxF,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,OAAO,C;K;4FAKX,Y;MAGI,QAAQ,E;MACR,OAAO,C;K;oFAKX,Y;MAGI,QAAQ,E;MACR,OAAO,C;K;8FAKX,Y;MAGI,QAAQ,E;MACR,OAAO,C;K;kGASX,oB;MAE8B,wB;QAAA,WAAkC,S;MAC5D,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;4FAUmE,qB;MAAQ,OAAO,M;K;8FAEd,qB;MAAQ,OAAQ,O;K;4FASrB,qB;MAAQ,OAAO,M;K;0GAER,qB;MAAQ,OAAc,a;K;8FAE7B,qB;MAAQ,OAAO,M;K;gGAEd,qB;MAAQ,OAAQ,O;K;8FASjB,qB;MAAQ,OAAO,M;K;gHAEL,qB;MAAQ,OAAiB,gB;K;wGASrC,qB;MAAQ,OAAa,Y;K;0GAEpB,qB;MAAQ,OAAc,a;K;wGAEvB,qB;MAAQ,OAAa,Y;K;oFCroB7F,4B;MAE6E,iBAAY,KAAZ,C;K;iGASnB,qB;MAAQ,OAAQ,Q;K;6FAEnB,qB;MAAQ,OAAO,M;K;+FAEd,qB;MAAQ,OAAQ,O;K;iGASF,qB;MAAQ,OAAU,S;K;+FAEnB,qB;MAAQ,OAAQ,Q;K;mGAS3B,qB;MAAQ,OAAW,U;K;mGAEnB,qB;MAAQ,OAAW,U;K;6GC1D/E,mb;MAEmC,yB;QAAA,YAAkB,C;MAAG,qB;QAAA,QAAiB,G;MAAK,sB;QAAA,SAAkB,G;MAAK,wB;QAAA,WAAmB,G;MAAI,kC;QAAA,qBAA6B,G;MAAI,qB;QAAA,QAAc,C;MAAG,qB;QAAA,QAAc,C;MAAG,qB;QAAA,QAAc,C;MAAG,2B;QAAA,cAAuB,E;MAAI,yB;QAAA,YAAsB,K;MAAO,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,sB;QAAA,SAAiB,C;MAAG,uB;QAAA,UAAkB,C;MAAG,6B;QAAA,gBAA8B,I;MAAM,sB;QAAA,SAAkB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SAAc,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACiB,QAAQ,E;MACR,EAAE,WAAF,IAAiB,S;MACjB,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,oBAAF,IAA0B,kB;MACiB,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,eAAF,IAAqB,a;MACrB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YA

AF,IAAkB,U;MACIB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;M  
ACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAA  
E,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O  
;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;6GC1BX,0C;MAEWc,oB;  
QAAA,OAAiB,I;MAAM,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,uB;QAAA,UAAoB,K;MA  
CpI,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,E  
AAE,SAAF,IAAe,O;MACf,OAAO,C;K;4EAmIX,4B;MAEkE,iBAAY,KAAZ,C;K;4EAEIE,qC;MAE2E,UAYY,K  
AAZ,IAAqB,O;K;4EAIbHG,4B;MAEuE,iBAAY,KAAZ,C;K;4EAEvE,qC;MAE+E,UAYY,KAAZ,IAAqB,O;K;4E  
AiBpG,4B;MAEuE,iBAAY,KAAZ,C;K;4EAEvE,qC;MAE+E,UAYY,KAAZ,IAAqB,O;K;4EAIgP,4B;MAEoE,i  
BAAY,KAAZ,C;K;2EAEpE,qC;MAE4E,UAYY,KAAZ,IAAqB,O;K;4EAKcjG,4B;MAE6E,iBAAY,KAAZ,C;K;4E  
AE7E,qC;MAEqF,UAYY,KAAZ,IAAqB,O;K;4EAgP1G,4B;MAEqE,iBAAY,KAAZ,C;K;4EAErE,qC;MAE6E,U  
AYY,KAAZ,IAAqB,O;K;uFJ57BIG,+H;MAE8B,sB;QAAA,SAAkB,S;MAAW,uB;QAAA,UAAmB,S;MAAW,oB;  
QAAA,OAAgB,S;MAAW,wB;QAAA,WAAoB,S;MAAW,8B;QAAA,iBAA0B,S;MAAW,oB;QAAA,OAAqB,S;M  
AAW,2B;QAAA,cAAmC,S;MAAW,qB;QAAA,QAAuB,S;MAAW,wB;QAAA,WAA6B,S;MAAW,yB;QAAA,YA  
AqB,S;MAAW,yB;QAAA,YAAsB,S;MAAW,wB;QAAA,WAAe,S;MAC5Z,QAAQ,E;MACR,EAAE,QAAF,IAAc  
,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,MAAF,IAAY,I;MACZ,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,g  
BAAF,IAAsB,c;MACtB,EAAE,MAAF,IAAY,I;MACZ,EAAE,aAAF,IAAmB,W;MACnB,EAAE,OAAF,IAAa,K;  
MACb,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,  
QAAF,IAAc,Q;MACd,OAAO,C;K;yFA0CX,uC;MAE+B,sB;QAAA,SAAiB,G;MAAK,0B;QAAA,aAAsB,I;MAA  
M,uB;QAAA,UAAmB,S;MACHG,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MACI  
B,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;qFAUgD,qB;MAAQ,OAAQ,E;K;mFAEX,qB;MAAQ,OAAQ,O;K;iF  
AEjB,qB;MAAQ,OAAO,M;K;mFAEd,qB;MAAQ,OAAQ,O;K;qFAEf,qB;MAAQ,OAAS,Q;K;mFAEIB,qB;MAA  
Q,OAAQ,O;K;mFAEhB,qB;MAAQ,OAAQ,O;K;mFAEhB,qB;MAAQ,OAAQ,O;K;qFASF,qB;MAAQ,OAAQ,E;K  
;yFAER,qB;MAAQ,OAAW,U;K;mFAEtB,qB;MAAQ,OAAQ,O;K;mFAEjB,qB;MAAQ,OAAO,M;K;qFAEd,qB;  
MAAQ,OAAQ,O;K;yFAEb,qB;MAAQ,OAAW,U;K;mFAEtB,qB;MAAQ,OAAQ,O;K;qFAEf,qB;MAAQ,OAAS,  
Q;K;qFAEjB,qB;MAAQ,OAAS,Q;K;uFAEjB,qB;MAAQ,OAAS,Q;K;mGAEV,qB;MAAQ,OAAgB,e;K;iGAEzB,q  
B;MAAQ,OAAe,c;K;qFAE9B,qB;MAAQ,OAAQ,O;K;qFAEf,qB;MAAQ,OAAS,Q;K;iFAEnB,qB;MAAQ,OAAO,  
M;K;yFASzB,qB;MAAQ,OAAW,U;K;+FAEhB,qB;MAAQ,OAAc,a;K;uFAE1B,qB;MAAQ,OAAU,S;K;iFAErB,q  
B;MAAQ,OAAO,M;K;iFASD,qB;MAAQ,OAAO,M;K;iGAER,qB;MAAQ,OAAc,a;K;uFAE1B,qB;MAAQ,OAAU  
,S;K;yFAS9B,qB;MAAQ,OAAU,S;K;yFAEjB,qB;MAAQ,OAAW,U;K;qFAErB,qB;MAAQ,OAAS,Q;K;yFAEf,qB  
;MAAQ,OAAW,U;K;+FAEhB,qB;MAAQ,OAAc,a;K;qGAEnB,qB;MAAQ,OAAiB,gB;K;qFAS3B,qB;MAAQ,OA  
AS,Q;K;mFAEIB,qB;MAAQ,OAAQ,O;K;uFAEf,qB;MAAQ,OAAS,Q;K;mFASxB,qB;MAAQ,OAAQ,O;K;mFAE  
jB,qB;MAAQ,OAAO,M;K;yFAEZ,qB;MAAQ,OAAU,S;K;qFAEpB,qB;MAAQ,OAAQ,O;K;qFAEf,qB;MAAQ,O  
AAS,Q;K;qGAET,qB;MAAQ,OAAiB,gB;K;+FKnR/F,gB;MAEkC,oB;QAAA,OAAgB,E;MAC9C,QAAQ,E;MAC  
R,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;+FAiBX,8B;MAEkC,4B;QAAA,eAAqB,S;MAAW,oB;QAAA,OAA  
gB,E;MAC9E,QAAQ,E;MACR,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;0EA  
UX,4B;MAE6D,iBAAY,KAAZ,C;K;+GC6B7D,sJ;MAEsC,mB;QAAA,MA4GuD,M;MA5GG,oB;QAAA,OAAgB  
,E;MAAI,oB;QAAA,OAAgB,E;MAAI,mB;QAAA,MAAe,E;MAAI,qB;QAAA,QAAiB,S;MAAW,oB;QAAA,OAA  
gB,S;MAAW,qB;QAAA,QAAiB,S;MAAW,qB;QAAA,QAAiB,S;MAAW,uB;QAAA,UAAmB,S;MAAW,yB;QA  
AA,YAAqB,S;MAAW,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,wB;QAAA,WAAqB,K;MA  
AO,kC;QAAA,qBAA+B,K;MAAO,sB;QAAA,SAAmB,K;MAAO,oB;QAAA,OAAa,I;MAAM,uB;QAAA,UAAc,  
E;MAC/gB,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE,MAAF,IAAY,I;MACZ,EAAE,MAAF,IAAY,  
I;MACZ,EAAE,KAAF,IAAW,G;MACX,EAAE,OAAF,IAAa,K;MACb,EAAE,MAAF,IAAY,I;MACZ,EAAE,OA  
AF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,WAAF,IAAiB,S;MACjB,E  
AAE,UAAF,IAAgB,Q;MACHB,EAAE,QAAF,IAAc,M;MACd,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,oBAAF,I  
AA0B,kB;MAC1B,EAAE,QAAF,IAAc,M;MACd,EAAE,MAAF,IAAY,I;MACZ,EAAE,SAAF,IAAe,O;MACf,OA  
AO,C;K;6GAWX,+B;MAEsE,oB;QAAA,OAAgB,S;MACIF,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAA  
E,OAAF,IAAa,K;MACb,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;qHASX,e;MAEyC,mB;QAAA,MAAe,E;MA



CpD,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,OAAO,C;K;mHAYBX,+D;MAEqE,sB;QAAA,SAAkB,E;M  
AAI,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACrK,QAAQ,E;MAC  
R,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAA  
kB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;iGAUwE,qB;MAAQ,OAAU,S;K;6FAEnB,qB;MAA  
Q,OAAS,Q;K;+FAEhB,qB;MAAQ,OAAU,S;K;2FASvB,qB;MAAQ,OAAO,M;K;yFAEhB,qB;MAAQ,OAAM,K;  
K;yFAEd,qB;MAAQ,OAAM,K;K;yGCrJ3F,uB;MAEsC,qB;QAAA,QAAiB,S;MAAW,oB;QAAA,ORy9MW,S;M  
Qx9MzE,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;6HAuCX,mF  
;MAEgD,oB;QAAA,OAAa,S;MAAW,sB;QAAA,SAAkB,S;MAAW,2B;QAAA,cAAuB,S;MAAW,sB;QAAA,SAA  
2C,S;MAAW,qB;QAAA,QAA6B,S;MAAW,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QA  
AA,WAAqB,K;MAC/S,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,aA  
AF,IAAmB,W;MACnB,EAAE,QAAF,IAAc,M;MACd,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf  
,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;uGA2DX,qC;MAEqC,mC;QAAA,  
sBAAgC,K;MAAO,oB;QAAA,OA4UD,Q;MA3UvE,QAAQ,E;MACR,EAAE,qBAAF,IAA2B,mB;MAC3B,EAAE  
,MAAF,IAAY,I;MACZ,OAAO,C;K;yGAmBX,yC;MAEsC,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;M  
AAO,wB;QAAA,WAAqB,K;MACHh,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MA  
CIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;yGAsBX,2B;MAGI,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;M  
ACd,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;+FA8BX,sE;MAEoD,wB;QAAA,WAAoB,I;MAAM,wB;QAAA,W  
AAqB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACpL,QA  
AQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,UAAF,IAAgB,Q;MACHb,EA  
AE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;6GAuB  
X,0D;MAE2D,sB;QAAA,SAAkB,M;MAAQ,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QA  
AA,WAAqB,K;MAC/J,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,SAA  
F,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;2GAaX,qC;MAE  
4D,sB;QAAA,SAAkB,S;MAAW,uB;QAAA,UAA0B,S;MAC/G,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MACH  
B,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;uHAuCX,mF;MAE6C,oB;QAAA,OAA  
a,S;MAAW,sB;QAAA,SAAkB,S;MAAW,2B;QAAA,cAAuB,S;MAAW,sB;QAAA,SAAmD,S;MAAW,qB;QAAA,  
QAA6B,S;MAAW,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACpT,Q  
AAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,E  
AAE,QAAF,IAAc,M;MACd,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;  
MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;qGA+BX,6D;MAEoC,4B;QAAA,eAAyB,K;MAAO,4B;QA  
AA,eAAyB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,yB;QAAA,YAAqB,S;MACnJ,QAAQ,E;MACR,EAAE,cAAF,  
IAAoB,Y;MACpB,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,WAAF,IAAiB,S;MA  
CjB,OAAO,C;K;yGakBX,4C;MAEsC,oB;QAAA,OAAGB,S;MAAW,uB;QAAA,UAAoB,S;MAAW,wB;QAAA,  
WAAsB,S;MAAW,uB;QAAA,UAA8B,S;MAC3J,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,SAAF,I  
AAe,O;MACf,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;+FAkCmE,qB;MAAQ,  
OAAa,Y;K;6FAEtB,qB;MAAQ,OAAY,W;K;+FAEnB,qB;MAAQ,OAAa,Y;K;6FAEtB,qB;MAAQ,OAAY,W;K;6F  
AEpB,qB;MAAQ,OAAY,W;K;6FAStC,qB;MAAQ,OAAY,W;K;6FAEpB,qB;MAAQ,OAAY,W;K;uFAEvB,qB;M  
AAQ,OAAS,Q;K;qFAEnB,qB;MAAQ,OAAO,M;K;uFASX,qB;MAAQ,OAAS,Q;K;yFAEjB,qB;MAAQ,OAAS,Q;  
K;qGAEX,qB;MAAQ,OAe,c;K;iFAEhC,qB;MAAQ,OAAM,K;K;iGCharE,0E;MAEoC,gC;QAAA,mBAA6B,K;  
MAAO,sB;QAAA,SAAkB,C;MAAG,qB;QAAA,QAAiB,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAA  
uB,K;MAAO,wB;QAAA,WAAqB,K;MAC3L,QAAQ,E;MACR,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,QAAF  
,IAAc,M;MACd,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EA  
AE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;mFAU8E,qB;MAAQ,OAAG,E;K;+FAEL,qB;MAAQ,OAac,a;K;iFAE7  
B,qB;MAAQ,OAAO,M;K;yFAEX,qB;MAAQ,OAaw,U;K;+EAEvB,qB;MAAQ,OAAO,M;K;+EAEf,qB;MAAQ,  
OAAO,M;K;oEIJlvG,yB;MAAA,kF;MAAA,0B;MAAA,uB;QAaI,IAAI,OAAO,CAAP,IAA8B,OAAO,KAAzC,C;  
UACI,MAAM,8BAAYB,wBAAqB,IAA9C,C;QAEV,OAAY,OAAL,IAAK,C;O;KAhBhB,C;0EAyCiC,qB;MAAQ,  
OAAA,SAAK,I;K;ImIrBV,6B;MAAC,qB;QAAA,8C;MAAA,kB;K;IACjC,2C;MAAA,e;MAAA,iB;MAAA,uB;K;I  
AAA,yC;MAAA,4C;O;MAKI,0E;MAEA,sE;K;IAFA,kD;MAAA,+B;MAAA,0C;K;IAEA,gD;MAAA,+B;MAAA

,wC;K;;IAPJ,qC;MAAA,yF;K;;IAAA,0C;MAAA,a;AAA,S;UAAA,+C;AAA,O;UAAA,6C;;UAAA,8D;;K;;IAyB  
mC,sC;MACnC,8B;K;;IAMqC,sC;MACrC,8B;K;;IC1DJ,iC;K;;ICMA,4B;K;;IA6BA,gD;K;;IC5BA,qC;K;;IAyBA,  
+B;K;;ICRqC,uC;MACjC,uB;QAAA,UAAaB,E;MACTb,qB;QAAA,+C;MADA,sB;MACA,kB;K;IAEA,4C;MAAA  
,e;MAAA,iB;MAAA,uB;K;IAAA,0C;MAAA,6C;O;MAKI,4E;MAGA,wE;K;;IAHA,mD;MAAA,gC;MAAA,2C;K;  
;IAGA,iD;MAAA,gC;MAAA,yC;K;;IARJ,sC;MAAA,2F;K;;IAAA,2C;MAAA,a;AAA,S;UAAA,gD;aAAA,O;UA  
AA,8C;;UAAA,+D;;K;;IAsByB,4B;MACzB,8B;K;;ICzC4C,8B;K;kDAI5C,mB;MAA6D,c;;QjJ6rD7C,Q;QADhB,I  
AAI,mCAAsB,cAA1B,C;UAAqC,aAAO,K;UAAp,e;;QACrB,sB;QAAhB,OAAgB,cAAhB,C;UAAgB,2B;UAAM,I  
iJ7rD6C,OjJ6rD/B,SiJ7rD+B,UjJ6rD7C,C;YAAwB,aAAO,I;YAAP,e;;;QAC9C,aAAO,K;;MiJ9rDsD,iB;K;uDAE7  
D,oB;MACa,c;;QjJqqDG,Q;QADhB,IAAI,ciJpqDA,QjJqQDA,iBiJpqDA,QjJqQDsB,UAA1B,C;UAAqC,aAAO,I;U  
AAP,e;;QACrB,OiJrqDZ,QjJqqDY,W;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CiJrqDP,oBjJqqDkB,  
OiJrqDIB,CjJqqDG,C;YAAyB,aAAO,K;YAAP,e;;;QAC/C,aAAO,I;;MiJtqDH,iB;K;2CAEJ,Y;MAAkC,qBAAQ,C  
;K;IAEqB,qE;MAAA,qB;QAC3D,OAAI,OAAO,uBAAX,GAAiB,mBAAjB,GAA6C,SAAH,EAAG,C;O;K;4CADj  
D,Y;MAAkC,4BAAa,IAAb,EAAMB,GAAAnB,EAAwB,GAAxB,kBAA6B,wCAA7B,C;K;2CAIIC,Y;MAI4C,uBAA  
gB,IAAhB,C;K;mDAE5C,iB;MAI4D,yBAAgB,IAAhB,EAAsB,KAAtB,C;K;;IC/BhE,8B;MAAA,e;MAAA,iB;MA  
AA,uB;K;IAAA,4B;MAAA,+B;O;MACI,4C;MACA,kD;MACA,0C;MACA,8C;K;;IAHA,mC;MAAA,kB;MAAA,  
2B;K;;IACA,sC;MAAA,kB;MAAA,8B;K;;IACA,kC;MAAA,kB;MAAA,0B;K;;IACA,oC;MAAA,kB;MAAA,4B;  
K;;IAJJ,wB;MAAA,sH;K;;IAAA,6B;MAAA,a;AAA,O;UAAA,gC;aAAA,U;UAAA,mC;aAAA,M;UAAA,+B;aA  
AA,Q;UAAA,iC;;UAAA,6D;;K;;IAOA,4B;MAKI,mD;MACA,2BAA4B,I;K;yCAE5B,Y;MAEiB,IAAN,I;MxJUX,  
IAAI,EwJXQ,mDxJWR,CAAJ,C;QACI,cAda,qB;QAEb,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwJZC,QAAM,oB  
AAN,M;aACH,M;UAAc,Y;UAAAd,K;aACA,O;UAAe,W;UAAf,K;;UACQ,wC;UAHL,K;;MAAP,W;K;sCAOJ,Y;M  
AIW,Q;MAHP,IAAI,CAAC,cAAL,C;QAAgB,MAAM,6B;MACTb,mD;MAEA,OAAO,2F;K;4DAGX,Y;MACI,iD;  
MACA,kB;MACA,OAAO,kD;K;+CAeX,iB;MAII,2BAA Y,K;MACZ,gD;K;sCAGJ,Y;MAII,+C;K;;ICjDkC,wB;M  
AoFtC,oC;MApFgE,6B;K;sCAIhE,Y;MAAuC,0C;K;2CAEvC,mB;MAAwD,uB;;QnJoU3C,Q;QADb,YAA Y,C;QA  
CC,sB;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,ImJrUmE,OnJqUrD,ImJrUqD,UnJqUnE,C;YACI,sBAAO,K;YAA  
P,wB;;UACJ,qB;;QAEJ,sBAAO,E;;;MmJzUiD,0B;K;+CAExD,mB;MAA4D,sB;;QnJ6V5D,eAAoB,0BAAa,SAAb,  
C;QACpB,OAAO,QAAS,cAAhB,C;UACI,ImJ/VsE,OnJ+VxD,QAAS,WmJ/V+C,UnJ+VtE,C;YACI,qBAAO,QAA  
S,Y;YAAhB,uB;;;QAGR,qBAAO,E;;;MmJnWqD,yB;K;0CAE5D,Y;MAA+C,+CAAiB,CAAjB,C;K;kDAE/C,iB;M  
AAyD,+CAAiB,KAAjB,C;K;6CAEzD,8B;MAA8D,gCAAQ,IAAR,EAAC,SAAd,EAAYB,OAAzB,C;K;IAEIC,wD;  
MAAgF,uB;MAA/E,kB;MAAMC,4B;MAC5D,eAAyB,C;MAGrB,+DAAkB,gBAAIB,EAA6B,OAA7B,EAAsC,W  
AAK,KAA3C,C;MACA,eAAa,UAAU,gBAAV,I;K;iDAGjB,iB;MACI,+DAAkB,KAAIB,EAAYB,YAAzB,C;MAE  
A,OAAO,wBAAK,mBAA Y,KAAZ,IAAL,C;K;4FAGY,Y;MAAQ,mB;K;;oCAGnC,iB;MAMI,IAAI,UAAU,IAAd,  
C;QAAoB,OAAO,I;MAC3B,IAAI,2BAAJ,C;QAAuB,OAAO,K;MAE9B,OAAO,2DAAC,IAAd,EAAoB,KAApB,C  
;K;sCAGX,Y;MAG+B,oEAAgB,IAAhB,C;K;IAE/B,2C;MAAA,oB;MACI,eACsB,C;K;kDAEtB,Y;MAAkC,sBAA  
Q,gB;K;+CAE1C,Y;MAEe,gB;MADX,IAAI,CAAC,cAAL,C;QAAgB,MAAM,6B;MACX,iE;MAAX,OAAO,+B;K  
;;IAO0B,sD;MAHzC,oB;MAGwD,iD;MAGhD,gEAAmB,KAAAnB,EAA0B,WAAkB,KAA5C,C;MACA,eAAa,K;K  
;0DAGjB,Y;MAAsC,sBAAQ,C;K;wDAE9C,Y;MAAgC,mB;K;uDAEhC,Y;MACI,IAAI,CAAC,kBAAL,C;QAAo  
B,MAAM,6B;MAC1B,OAAO,yBAAI,mCAAJ,EAAL,YAAJ,E;K;4DAGX,Y;MAAoC,sBAAQ,CAAR,I;K;;IAGxC,  
kC;MAAA,sC;K;iEACI,uB;MACI,IAAI,QAAQ,CAAR,IAAa,SAAS,IAA1B,C;QACI,MAAM,8BAA0B,YAAS,K  
AAT,gBAAuB,IAAjD,C;;K;kEAIId,uB;MACI,IAAI,QAAQ,CAAR,IAAa,QAAQ,IAAzB,C;QACI,MAAM,8BAA0  
B,YAAS,KAAAT,gBAAuB,IAAjD,C;;K;iEAIId,oC;MACI,IAAI,YAA Y,CAAZ,IAAiB,UAAU,IAA/B,C;QACI,MAA  
M,8BAA0B,gBAAa,SAAb,mBAAkC,OAAIC,gBAAkD,IAA5E,C;;MAEV,IAAI,YAA Y,OAAhB,C;QACI,MAAM,  
gCAAyB,gBAAa,SAAb,oBAAmC,OAA5D,C;;K;kEAIId,sC;MACI,IAAI,aAAa,CAAb,IAAkB,WAAW,IAAjC,C;Q  
ACI,MAAM,8BAA0B,iBAAC,UAAAd,oBAAqC,QAArC,gBAAAsD,IAAhF,C;;MAEV,IAAI,aAAa,QAAjB,C;QACI,  
MAAM,gCAAyB,iBAAC,UAAAd,qBAAsC,QAA/D,C;;K;+DAId,a;MAEc,UACsB,M;MAFhC,iBAAe,C;MACL,mB  
;MAAV,OAAU,cAAV,C;QAAU,mB;QACN,aAAW,MAAK,UAAAL,SAAiB,6DAAiB,CAAIC,K;;MAEf,OAAO,U;  
K;6DAGX,oB;MAIiB,Q;MAHb,IAAI,CAAE,KAAF,KAAU,KAAM,KAApB,C;QAA0B,OAAO,K;MAEjC,oBAA  
oB,KAAAM,W;MACb,mB;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,gBAAgB,aAAc,O;QAC9B,IAAI,cAAQ,SAAR,  
CAAJ,C;UACI,OAAO,K;;MAGf,OAAO,I;K;;IAjDf,8C;MAAA,6C;QAAA,4B;;MAAA,sC;K;;ICnFwC,uB;MAY

HxC,mC;MAzCA,uBAC6B,I;MAMC7B,yBACsC,I;K;8CAnHtC,e;MACI,OAAO,6BAAC,GAAd,S;K;gDAGX,iB;MAAwE,gBAAR,Y;MAAQ,c;;QpJorDxD,Q;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,aAAO,K;UAAP,e;;QA CrB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IoJprDwD,OpJorD1C,OoJprD6C,MAAH,QpJorDxD,C;Y AAwB,aAAO,I;YAAP,e;;;QAC9C,aAAO,K;;;MoJrrDyD,iB;K;kDAEhE,iB;MAEI,IAAI,gCAAJ,C;QAA+B,OAAO ,K;MACtC,UAAU,KAAM,I;MACHB,YAAY,KAAM,M;MnKmNO,Q;MmKINzB,enKkN4C,CAAnB,mDAAmB,Y mKINzB,GnKkNyB,C;MmKhN5C,IAAI,eAAS,QAAT,CAAJ,C;QACI,OAAO,K;;;MAIP,6B;MAAA,W;QnK4NqB, U;QmK5ND,UnK4NoB,CAAnB,uDAAmB,oBmK5NP,GnK4NO,C;;MmK5N5C,W;QACI,OAAO,K;;;MAGX,OAA O,I;K;mCAIX,iB;MAMI,IAAI,UAAU,IAAd,C;QAAoB,OAAO,I;MAC3B,IAAI,0BAAJ,C;QAAyB,OAAO,K;MA ChC,IAAI,cAAQ,KAAM,KAA1B,C;QAAwB,OAAO,K;MAEV,gBAAd,KAAM,Q;MAAQ,c;;QpJ+nDT,Q;QADhB ,IAAI,wCAAsB,mBAA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAA M,IAAI,CoJ/nDK,2BpJ+nDM,OoJ/nDN,CpJ+nDT,C;YAAyB,aAAO,K;YAAP,e;;;QAC/C,aAAO,I;;;MoJhoDH,iB; K;sCAGJ,e;MAAwC,Q;MAAA,4CAAc,GAAd,8B;K;qCAGxC,Y;MAK+B,OAAQ,SAAR,YAAQ,C;K;oCAEvC,Y; MAAkC,qBAAQ,C;K;mFACnB,Y;MAAQ,OAAA,YAAQ,K;K;IAWnB,0E;MAAA,wC;MAAS,sB;K;8EACb,mB; MAAsD,+CAAY,OAAZ,C;K;IAI3C,sG;MAAA,kD;K;8FACH,Y;MAAkC,OAAA,0BAAc,U;K;2FACHd,Y;MAAy B,OAAA,0BAAc,OAAO,I;K;;wEAJtD,Y;MACI,oBAAoB,6BAAQ,W;MAC5B,+F;K;sHAMmB,Y;MAAQ,OAAA, qBAAiB,K;K;;mFAB5D,Y;MACI,IAAI,4BAAJ,C;QACI,+E;;MacJ,OAAO,mC;K;IAOwD,uD;MAAA,qB;QAAE,2 CAAS,EAAT,C;O;K;qCAAzE,Y;MAAkC,OAAQ,eAAR,YAAQ,EAAa,IAAb,EAAMB,GAAAnB,EAAwB,GAAxB, kBAA6B,iCAA7B,C;K;+CAE1C,iB;MAAuD,+BAAS,KAAM,IAAf,IAAsB,GAAtB,GAA4B,wBAAS,KAAM,MA Af,C;K;+CAEnF,a;MAAwC,OAAI,MAAM,IAAV,GAAGB,YAAhB,GAAoC,SAAF,CAAE,C;K;IAWtD,4E;MAA A,wC;MAAS,6B;K;gFACf,mB;MAAsE,iDAAc,OAAc,C;K;IAI3D,wG;MAAA,kD;K;gGACH,Y;MAAkC,OAAA, 0BAAc,U;K;6FACHd,Y;MAAyB,OAAA,0BAAc,OAAO,M;K;;0EAJtD,Y;MACI,oBAAoB,6BAAQ,W;MAC5B,i G;K;wHAMmB,Y;MAAQ,OAAA,qBAAiB,K;K;;qFAB5D,Y;MACI,IAAI,8BAAJ,C;QACI,mF;;MacJ,OAAO,qC; K;oDAMf,e;MAA8D,gBAAR,Y;MAAQ,sB;;QpJmJ9C,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAA M,IoJnJsD,OpJmJxC,OoJnJ2C,IAAH,MpJmJtD,C;YAAwB,qBAAO,O;YAAP,uB;;QAC9C,qBAAO,I;;;MoJpJ+C, yB;K;IAEtD,iC;MAAA,qC;K;4DAEI,a;MAAiE,gC;MAAX,OAAU,CAAC,kBAAN,CAAM,0DAAmB,CAApB,KA A4B,oBAAjC,CAAIc,8DAAqB,CAAjD,C;K;4DACHe,a;MAAyD,OAAU,SAAL,CAAO,IAAF,mBAAL,CAAY,M AAP,C;K;0DACnE,oB;MACI,IAAI,gCAAJ,C;QAA+B,OAAO,K;MACtC,OAAO,OAAA,CAAE,IAAF,EAAS,KA AM,IAAf,KAAsB,OAAA,CAAE,MAAF,EAAW,KAAM,MAAJB,C;K;;IANrC,6C;MAAA,4C;QAAA,2B;;MAAA ,qC;K;;IChIqC,uB;MAkBrC,mC;MAIB+D,6B;K;mCAE/D,iB;MAMI,IAAI,UAAU,IAAd,C;QAAoB,OAAO,I;MA C3B,IAAI,0BAAJ,C;QAAsB,OAAO,K;MAC7B,OAAO,sDAAU,IAAV,EAAGB,KAAhB,C;K;qCAGX,Y;MAG+B, qEAAkB,IAAIB,C;K;IAE/B,iC;MAAA,qC;K;gEACI,a;MAEoB,Q;MADhB,iBAAE,C;MACC,mB;MAAhB,OAAg B,cAAhB,C;QAAgB,yB;QACC,U;QAAb,2BAAa,yEAAuB,CAApC,K;;MAEJ,OAAO,U;K;wDAGX,oB;MACI,IA AI,CAAE,KAAP,KAAM,KAAPB,C;QAA0B,OAAO,K;MACjC,OAAO,CrK8OsG,qBqK9OxF,KrK8OwF, C;K;;IqKzPrH,6C;MAAA,4C;QAAA,2B;;MAAA,qC;K;;MCghBA,kC;MA9hBA,cAAwB,C;MACxB,yB;MAEA, sBAAyB,C;;kFAAzB,Y;MAAA,0B;K,OAAA,gB;MAAA,0B;K;4CA8BA,uB;MAOI,IAAI,cAAc,CAAIB,C;QAAq B,MAAM,6BAAsB,mBAAtB,C;MAC3B,IAAI,eAAe,kBAAy,OAA/B,C;QAAqC,M;MACrC,IAAI,uBAAgB,qDA ApB,C;QACI,qBAAc,gBAAyB,gBAAZ,WAAy,EAAC,EAAd,CAAZB,O;QACd,M;;MAGJ,kBAAkB,uDAAY,kB AAY,OAAxB,EAA8B,WAA9B,C;MACIB,oBAAa,WAAb,C;K;0CAGJ,uB;MAII,kBAAkB,gBAAmB,WAAAnB,O; M3J20BtB,U2J10BI,kB3J00BJ,E2J10ByB,W3J00BzB,E2J10BsC,C3J00BtC,E2J10ByC,W3J00BzC,E2J10B+C,kB AAY,O3J00B3D,C;MAAA,U2Jz0BI,kB3Jy0BJ,E2Jz0ByB,W3Jy0BzB,E2Jz0BsC,kBAAy,OAAZ,GAAmB,WAAAn B,I3Jy0BtC,E2Jz0B+D,C3Jy0B/D,E2Jz0BkE,W3Jy0BIE,C;M2Jx0BI,cAAO,C;MACP,qBAAc,W;K;yCAGIB,yB;M AGW,Q;MAAP,OAAO,2BAAy,aAAZ,4D;K;yCAGX,iB;MAA2C,OAAI,SAAS,kBAAy,OAAzB,GAA+B,QAAQ, kBAAy,OAApB,IAA/B,GAA6D,K;K;yCAExG,iB;MAA2C,OAAI,QAAQ,CAAZ,GAAe,QAAQ,kBAAy,OAApB, IAAf,GAA6C,K;K;2CAExF,iB;MACoD,0BAAy,cAAO,KAAP,IAAZ,C;K;yCAEpD,iB;MAA2C,OAAI,UAAqB,c AAZ,kBAAy,CAAZB,GAAoC,CAApC,GAA2C,QAAQ,CAAR,I;K;yCAEtF,iB;MAA2C,OAAI,UAAS,CAAb,GA A4B,cAAZ,kBAAy,CAA5B,GAA2C,QAAQ,CAAR,I;K;mCAEtF,Y;MAAkC,qBAAQ,C;K;iCAE1C,Y;MAGwB,I AAI,cAAJ,C;QAAe,MAAM,2BAAuB,sBAAvB,C;;QAnBIC,Q;QAmBa,OAnBb,2BAmBkG,WAnBIG,4D;;K;uCA qBX,Y;MAG+B,Q;MAAA,IAAI,cAAJ,C;QAAA,OAAe,I;;QAxBnC,U;QAwBoB,OAxBpB,6BAwByD,WAxBzD,g

E;;MAwBoB,W;K;gCAE/B,Y;MAGuB,IAAI,cAAJ,C;QA Ae,MAAM,2BA AuB,sBA AvB,C;;QA7BjC,Q;QA6BY,O  
A7BZ,2BAQyC,mBAAY,cAqB0D,sBArB1D,IAAZ,CARzC,4D;;K;sCA+BX,Y;MAG8B,Q;MAAA,IAAI,cAAJ,C;  
QAAA,OAAe,I;;QALCIC,U;QAKCmB,OAlCnB,6BAQyC,mBAAY,cA0BiB,sBA1BjB,IAAZ,CARzC,gE;;MAkCm  
B,W;K;0CAE9B,mB;MAII,sBA Ae,YAAO,CAAP,IAAf,C;MAEA,cAAO,mBAAY,WAAZ,C;MACP,mBAAY,WA  
AZ,IAAoB,O;MACpB,wBAAQ,CAAR,I;K;yCAGJ,mB;MAII,sBA Ae,YAAO,CAAP,IAAf,C;MAEA,mBA7CgD,m  
BAAY,cA6CIC,SA7CkC,IAAZ,CA6ChD,IAAmC,O;MACnC,wBAAQ,CAAR,I;K;uCAGJ,Y;MAII,IAAI,cAAJ,C;  
QA Ae,MAAM,2BA AuB,sBA AvB,C;MA7Dd,Q;MA+DP,cA/DO,2BA+DmB,WA/DnB,4D;MAGeP,mBAAY,WA  
AZ,IAAoB,I;MACpB,cAAO,mBAAY,WAAZ,C;MACP,wBAAQ,CAAR,I;MACA,OAAO,O;K;6CAGX,Y;MAGq  
C,OAAI,cAAJ,GAAe,IAAf,GAAYb,kB;K;sCAE9D,Y;MAII,IAAI,cAAJ,C;QA Ae,MAAM,2BA AuB,sBA AvB,C;M  
AErB,wBAzEgD,mBAAY,cAyEtB,sBAzEsB,IAAZ,C;MARzC,Q;MAkFP,cAlFO,2BAkFmB,iBAIFnB,4D;MAMF  
P,mBAAY,iBAAZ,IAAiC,I;MACjC,wBAAQ,CAAR,I;MACA,OAAO,O;K;4CAGX,Y;MAGoC,OAAI,cAAJ,GAA  
e,IAAf,GAAYb,iB;K;qCAE7D,mB;MAEI,mBAAQ,OAAR,C;MACA,OAAO,I;K;uCAGX,0B;MACI,oCAAa,4BA  
AmB,KAA nB,EAA0B,SA A1B,C;MAEb,IAAI,UAAS,SAAb,C;QACI,mBAAQ,OAAR,C;QACA,M;aACG,IAAI,U  
AAS,CAAb,C;QACH,oBAAS,OAAT,C;QACA,M;;MAGJ,sBA Ae,YAAO,CAAP,IAAf,C;MA2BA,oBAjIgd,mBA  
AY,cAiI1B,KAJI0B,IAAZ,C;MAMlhD,IAAI,QAAS,SAAD,GAAQ,CAAR,IA Ae,CAA3B,C;QAEI,+BAA+B,mBA  
AY,aAAZ,C;QAC/B,sBAAsB,mBAAY,WAAZ,C;QAEtB,IAAI,4BAA4B,WAAhC,C;UACI,mBAAY,eAAZ,IAA+  
B,mBAAY,WAAZ,C;U3JgrB3C,U2J/qBY,kB3J+qBZ,E2J/qBiC,kB3J+qBjC,E2J/qB8C,W3J+qB9C,E2J/qBoD,cA  
AO,CAAP,I3J+qBpD,E2J/qB8D,2BAA2B,CAA3B,I3J+qB9D,C;;UAAA,U2J7qBY,kB3J6qBZ,E2J7qBiC,kB3J6qB  
jC,E2J7qB8C,cAAO,CAAP,I3J6qB9C,E2J7qBwD,W3J6qBxD,E2J7qB8D,kBAAY,O3J6qB1E,C;U2J5qBY,mBA  
AY,kBAAY,OAAZ,GAAmB,CAAnB,IAAZ,IAAoC,mBAAY,CAAZ,C;U3J4qBhD,U2J3qBY,kB3J2qBZ,E2J3qBiC,  
kB3J2qBjC,E2J3qB8C,C3J2qB9C,E2J3qBiD,C3J2qBjD,E2J3qBoD,2BAA2B,CAA3B,I3J2qBpD,C;;Q2JxqBQ,mB  
AAY,wBAAZ,IAAwC,O;QACxC,cAAO,e;;QAGP,WArJ4C,mBAAY,cAqJ/B,SArJ+B,IAAZ,C;QAU5C,IAAI,gB  
AAgB,IAApB,C;U3JkqBR,U2JjqBY,kB3JiqBZ,E2JjqBiC,kB3JiqBjC,E2JjqB8C,gBAAgB,CAAhB,I3JiqB9C,E2Jjq  
BiE,a3JiqBjE,E2JjqBgF,I3JiqBhF,C;;UAAA,U2J/pBY,kB3J+pBZ,E2J/pBiC,kB3J+pBjC,E2J/pB8C,C3J+pB9C,E2J  
/pBiD,C3J+pBjD,E2J/pBoD,I3J+pBpD,C;U2J9pBY,mBAAY,CAAZ,IAAiB,mBAAY,kBAAY,OAAZ,GAAmB,CA  
AnB,IAAZ,C;U3J8pB7B,U2J7pBY,kB3J6pBZ,E2J7pBiC,kB3J6pBjC,E2J7pB8C,gBAAgB,CAAhB,I3J6pB9C,E2J  
7pBiE,a3J6pBjE,E2J7pBgF,kBAAY,OAAZ,GAAmB,CAAnB,I3J6pBhF,C;;Q2J1pBQ,mBAAY,aAAZ,IAA6B,O;;  
MAEjC,wBAAQ,CAAR,I;K;oDAGJ,mC;MAGkD,UAIxB,M;MANTb,eAAe,QAAS,W;MAEsB,OAAZ,kBAAY,O;  
MAA9C,iBAAc,aAAd,wB;QACI,IAAI,CAAC,QAAS,UAA d,C;UAAyB,K;QACzB,mBAAY,KAAZ,IAAqB,QAA  
S,O;;MAEZ,oB;MAAtB,mBAAc,CAAd,8B;QACI,IAAI,CAAC,QAAS,UAA d,C;UAAyB,K;QACzB,mBAAY,OA  
AZ,IAAqB,QAAS,O;;MAGIc,wBAAQ,QAAS,KAAjB,I;K;0CAGJ,oB;MACI,IAAI,QAAS,UAA b,C;QAAwB,OA  
AO,K;MAC/B,sBA Ae,IAAK,KAAL,GAAY,QAAS,KAArB,IAAf,C;MACA,8BA tLgD,mBAAY,cAsLvB,SAtLuB,  
IAAZ,CAsLhD,EAA4C,QAA5C,C;MACA,OAAO,I;K;0CAGX,2B;MACI,oCAAa,4BAAmB,KAA nB,EAA0B,SA  
A1B,C;MAEb,IAAI,QAAS,UAA b,C;QACI,OAAO,K;aACJ,IAAI,UAAS,SAAb,C;QACH,OAAO,oBAAO,QAAP,  
C;;MAGX,sBA Ae,IAAK,KAAL,GAAY,QAAS,KAArB,IAAf,C;MAEA,WArMgD,mBAAY,cAqMnC,SArMmC,I  
AAZ,C;MAsMhD,oBA tMgD,mBAAY,cAsM1B,KAtM0B,IAAZ,C;MAuMhD,mBAAmB,QAAS,K;MAE5B,IAAI,  
QAAS,SAAD,GAAQ,CAAR,IA Ae,CAA3B,C;QAGI,kBAAkB,cAAO,YAAP,I;QAEIb,IAAI,iBA AiB,WAArB,C;U  
ACI,IAAI,eAAe,CAAnB,C;Y3J0mBZ,U2JzmBgB,kB3JymBhB,E2JzmBqC,kB3JymBrC,E2JzmBkD,W3JymBID,E  
2JzmB+D,W3JymB/D,E2JzmBqE,a3JymBrE,C;;Y2JvmbG,4BA Ae,kBAAY,OAA3B,I;YACA,sBAAsB,gBAAgB,  
WAAhB,I;YActB,kBAAkB,kBAAY,OAAZ,GAAmB,WAA nB,I;YAEIb,IAAI,eAAe,eAA nB,C;c3JmmBhB,U2Jlm  
BoB,kB3JkmBpB,E2JlmByC,kB3JkmBzC,E2JlmBsD,W3JkmBtD,E2JlmBmE,W3JkmBnE,E2JlmByE,a3JkmBzE,  
C;;cAAA,U2JhmBoB,kB3JgmBpB,E2JhmByC,kB3JgmBzC,E2JhmBsD,W3JgmBtD,E2JhmBmE,W3JgmBnE,E2Jh  
mByE,cAAO,WAAP,I3JgmBzE,C;cAAA,U2J/IBoB,kB3J+IBpB,E2J/IByC,kB3J+IBzC,E2J/IBsD,C3J+IBtD,E2J/IB  
yD,cAAO,WAAP,I3J+IBzD,E2J/IB6E,a3J+IB7E,C;;;UAAA,U2J3IBY,kB3J2IBZ,E2J3IBiC,kB3J2IBjC,E2J3IB8C,  
W3J2IB9C,E2J3IB2D,W3J2IB3D,E2J3IBiE,kBAAY,O3J2IB7E,C;U2J1IBY,IAAI,gBAAgB,aAApB,C;Y3J0IBZ,U2  
JzIBgB,kB3JyIBhB,E2JzIBqC,kB3JyIBrC,E2JzIBkD,kBAAY,OAAZ,GAAmB,YAA nB,I3JyIBID,E2JzIBmF,C3JyI  
BnF,E2JzIBsF,a3JyIBtF,C;;YAAA,U2JvIBgB,kB3JulBhB,E2JvIBqC,kB3JulBrC,E2JvIBkD,kBAAY,OAAZ,GAAm  
B,YAA nB,I3JulBID,E2JvIBmF,C3JulBnF,E2JvIBsF,Y3JulBtF,C;YAAA,U2JtIBgB,kB3JslBhB,E2JtIBqC,kB3JslBr

C,E2JtlBkD,C3JslBID,E2JtlBqD,Y3JslBrD,E2JtlBmE,a3JslBnE,C;;;Q2JnlBQ,cAAO,W;QACP,8BAAuB,mBAAY,gBAAGB,YAAhB,IAAZ,CAAvB,EAaKE,QAAIE,C;;QAIA,2BAA2B,gBAAGB,YAAhB,I;QAE3B,IAAI,gBAAGB,IAApB,C;UACI,IAAI,QAAO,YAAP,SAAuB,kBAAY,OAAvC,C;Y3J2kBZ,U2J1kBGB,kB3J0kBhB,E2J1kBqC,kB3J0kBrC,E2J1kBkD,oB3J0kBID,E2J1kBwE,a3J0kBxE,E2J1kBuF,I3J0kBvF,C;;Y2JxkBGB,IAAI,wBAAwB,kBAAY,OAAx,C;c3JwkBhB,U2JvkBoB,kB3JukBpB,E2JvkByC,kB3JukBzC,E2JvkBsD,uBAAuB,kBAAY,OAAnc,I3JukBtD,E2JvkB+F,a3JukB/F,E2JvkB8G,I3JukB9G,C;;c2JrkBoB,mBAAmB,OAAO,YAAP,GAAaB,kBAAY,OAAIC,I;c3JqkBvC,U2JpkBoB,kB3JokBpB,E2JpkByC,kB3JokBzC,E2JpkBsD,C3JokBtD,E2JpkByD,OAAO,YAAP,I3JokBzD,E2JpkB8E,I3JokB9E,C;cAAA,U2JnkBoB,kB3JmkBpB,E2JnkByC,kB3JmkBzC,E2JnkBsD,oB3JmkBtD,E2JnkB4E,a3JmkB5E,E2JnkB2F,OAAO,YAAP,I3JmkB3F,C;;;UAAA,U2J/jBY,kB3J+jBZ,E2J/jBiC,kB3J+jBjC,E2J/jB8C,Y3J+jB9C,E2J/jB4D,C3J+jB5D,E2J/jB+D,I3J+jB/D,C;U2J9jBY,IAAI,wBAAwB,kBAAY,OAAx,C;Y3J8jBZ,U2J7jBgB,kB3J6jBhB,E2J7jBqC,kB3J6jBrC,E2J7jBkD,uBAAuB,kBAAY,OAAnc,I3J6jBID,E2J7jB2F,a3J6jB3F,E2J7jB0G,kBAAY,O3J6jBtH,C;;YAAA,U2J3jBgB,kB3J2jBhB,E2J3jBqC,kB3J2jBrC,E2J3jBkD,C3J2jBID,E2J3jBqD,kBAAY,OAAZ,GAAmB,YAAnc,I3J2jBrD,E2J3jBsF,kBAAY,O3J2jBIG,C;YAAA,U2J1jBgB,kB3J0jBhB,E2J1jBqC,kB3J0jBrC,E2J1jBkD,oB3J0jBID,E2J1jBwE,a3J0jBxE,E2J1jBuF,kBAAY,OAAZ,GAAmB,YAAnc,I3J0jBvF,C;;;Q2JvjBQ,8BAAuB,AAAvB,EAAsC,QAAIC,C;;MAGJ,OAAO,I;K;uCAGX,iB;MACI,oCAAa,2BAaKB,KAAIB,EAAYB,SAAZB,C;MAjRN,Q;MAmRP,OAnRO,2BAQyC,mBAAY,cA2Q3B,KA3Q2B,IAAZ,CARzC,4D;K;uCAaRX,0B;MACI,oCAAa,2BAaKB,KAAIB,EAAYB,SAAZB,C;MAEb,oBAjRgD,mBAAY,cAiR1B,KAjR0B,IAAZ,C;MARzC,Q;MA0RP,iBA1RO,2BA0RsB,aA1RtB,4D;MA2RP,mBAAY,aAAZ,IAA6B,O;MAE7B,OAAO,U;K;OCAGX,mB;MAAoD,0BAAQ,OAAR,MAAoB,E;K;yCAExE,mB;MAIsB,IAIA,IAJA,EAIB,M;MAPzC,WA3RgD,mBAAY,cA2RnC,SA3RmC,IAAZ,C;MA6RhD,IAAI,cAAO,IAAX,C;QACI,iBAAc,WAAAd,UAAyB,IAAZB,U;UACI,IAAI,gBAAW,mBAAY,KAAZ,CAAX,CAAJ,C;YAAmC,OAAO,QAAQ,WAAR,I;;aAE3C,IAAI,eAAQ,IAAZ,C;QACW,kB;QAAuB,SAAZ,kBAAY,O;QAAR,c,qD;UACI,IAAI,gBAAW,mBAAY,OAAZ,CAAX,CAAJ,C;YAAmC,OAAO,UAAQ,WAAR,I;;QAE9C,mBAAc,CAAd,YAAsB,IAAtB,Y;UACI,IAAI,gBAAW,mBAAY,OAAZ,CAAX,CAAJ,C;YAAmC,OAAO,UAAQ,kBAAY,OAApB,GAA2B,WAA3B,I;;MAIID,OAAO,E;K;6CAGX,mB;MAIsC,UAOJ,MAPI,EAoA,M;MAV/C,WA9SgD,mBAAY,cA8SnC,SA9SmC,IAAZ,C;MAGThD,IAAI,cAAO,IAAX,C;QACkC,kB;QAA9B,iBAAc,OAAO,CAAP,IAAd,yB;UACI,IAAI,gBAAW,mBAAY,KAAZ,CAAX,CAAJ,C;YAAmC,OAAO,QAAQ,WAAR,I;;aAE3C,IAAI,cAAO,IAAX,C;QACH,mBAAc,OAAO,CAAP,IAAd,aA8B,CA9B,Y;UACI,IAAI,gBAAW,mBAAY,OAAZ,CAAX,CAAJ,C;YAAmC,OAAO,UAAQ,kBAAY,OAApB,GAA2B,WAA3B,I;;QAEpB,uBAAZ,kBAAY,C;QAAiB,oB;QAA3C,wD;UACI,IAAI,gBAAW,mBAAY,OAAZ,CAAX,CAAJ,C;YAAmC,OAAO,UAAQ,WAAR,I;;MAIID,OAAO,E;K;wCAGX,mB;MACI,YAAY,mBAAQ,OAAR,C;MACZ,IAAI,UAAS,EAAb,C;QAAiB,OAAO,K;MACxB,sBAAS,KAAT,C;MACA,OAAO,I;K;4CAGX,iB;MACI,oCAAa,2BAaKB,KAAIB,EAAYB,SAAZB,C;MAEb,IAAI,UAAS,sBAAb,C;QACI,OAAO,iB;aACJ,IAAI,UAAS,CAAb,C;QACH,OAAO,kB;;MAGX,oBAhVgD,mBAAY,cAgV1B,KAhV0B,IAAZ,C;MARzC,Q;MAyVP,cAzVO,2BAyVmB,aAzVnB,4D;MA2VP,IAAI,QAAQ,aAAS,CAAR,B,C;QAEI,IAAI,iBAAiB,WAArB,C;U3JoeR,U2JneY,kB3JmeZ,E2JneiC,kB3JmejC,E2Jne8C,cAAO,CAAP,I3Jme9C,E2JnewD,W3JmexD,E2Jne8D,a3Jme9D,C;;UAAA,U2JjeY,kB3JjeZ,E2JjeiC,kB3JiejC,E2Jje8C,C3Jje9C,E2JjeiD,C3JiejD,E2JjeoD,a3JjepD,C;U2JheY,mBAAY,CAAZ,IAAiB,mBAAY,kBAAY,OAAZ,GAAmB,CAAnB,IAAZ,C;U3Jge7B,U2J/dY,kB3J+dZ,E2J/diC,kB3J+djC,E2J/d8C,cAAO,CAAP,I3J+d9C,E2J/dwD,W3J+dxD,E2J/d8D,kBAAY,OAAZ,GAAmB,CAAnB,I3J+d9D,C;;Q2J5dQ,mBAAY,WAAZ,IAAoB,I;QACpB,cAAO,mBAAY,WAAZ,C;;QAGP,wBAjW4C,mBAAY,cAiWIB,sBAjWkB,IAAZ,C;QAmW5C,IAAI,iBAAiB,iBAArB,C;U3JsdR,U2JrdY,kB3JqdZ,E2JrdiC,kB3JqdjC,E2Jrd8C,a3Jqd9C,E2Jrd6D,gBAAgB,CAAhB,I3Jqd7D,E2JrdgF,oBAAoB,CAApB,I3JqdhF,C;;UAAA,U2JndY,kB3JmdZ,E2JndiC,kB3JmdjC,E2Jnd8C,a3Jmd9C,E2Jnd6D,gBAAGB,CAAhB,I3Jmd7D,E2JndgF,kBAAY,O3Jmd5F,C;U2JldY,mBAAY,kBAAY,OAAZ,GAAmB,CAAnB,IAAZ,IAAoC,mBAAY,CAAZ,C;U3JkdhD,U2JjdY,kB3JidZ,E2JjdiC,kB3JidjC,E2Jjd8C,C3Jid9C,E2JjdiD,C3JidjD,E2JjdoD,oBAAoB,CAApB,I3JidpD,C;;Q2J9cQ,mBAAY,iBAAZ,IAAiC,I;;MAErC,wBAAQ,CAAR,I;MAEA,OAAO,O;K;6CAGX,oB;MAAKe,0B;;QAa5C,wD;QART,aAAL,IAAK,U;QAAL,Y;UAA8B,SAAZ,kB3KoxOnB,YAAQ,C;;Q2KpxOX,W;UACI,yBAAO,K;UAAP,2B;;QAEJ,WA1XgD,mBAAY,cA0XnC,SA1XmC,IAAZ,C;QA2XhD,cAAc,W;QACd,eAAe,K;QAEf,IAAI,cAAO,IAAX,C;UACI,iBAAc,WAAAd,UAAyB,IAAZB,U;YACI,cAAc,mBAAY,KAAZ,C;YAGd,IAjBsE,CAAU,wBAiBIE,0EAjBkE,CAiBhF,C;cACI,mBAAY,gBAAZ,EA

AY,wBAAZ,YAAyB,O;;cAEzB,WAAW,I;;UAGP,OAAZ,kBAAY,EAAK,IAAL,EAAW,OAAx,EAAoB,IAApB,C;;UAGE,oB;UAAuB,SAAZ,kBAAY,O;UAArC,uD;YACI,gBAAC,mBAAY,OAAZ,C;YACd,mBAAY,OAAZ,IAAqB,I;YAGrB,IA/BsE,CAAU,wBA+BIE,kFA/BkE,CA+BhF,C;cACI,mBAAY,gBAAZ,EAAy,wBAAZ,YAAyB,S;;cAEzB,WAAW,I;;UAGnB,UAAU,mBAAY,OAAZ,C;UAEV,mBAAC,CAAd,YAAsB,IAAtB,Y;YACI,gBAAC,mBAAY,OAAZ,C;YACd,mBAAY,OAAZ,IAAqB,I;YAGrB,IA5CsE,CAAU,wBA4CIE,kFA5CkE,CA4ChF,C;cACI,mBAAY,OAAZ,IAAuB,S;cACvB,UAAU,mBAAY,OAAZ,C;;cAEV,WAAW,I;;;QAIvB,IAAI,QAAJ,C;UACI,YAAO,mBAAY,UAAU,WAAV,IAAZ,C;QAEX,yBAAO,Q;;;MAvDuD,6B;K;6CAEIE,oB;MAAkE,0B;;QAW5C,wD;QART,aAAL,IAAK,U;QAAL,Y;UAA8B,SAAZ,kB3KoxOnB,YAAQ,C;;Q2KpxOX,W;UACI,yBAAO,K;UAAP,2B;;QAEJ,WA1XgD,mBAAY,cA0XnC,SA1XmC,IAAZ,C;QA2XhD,cAAc,W;QACd,eAAe,K;QAEf,IAAI,cAAO,IAAX,C;UACI,iBAAC,WAAAd,UAAyB,IAAZB,U;YACI,cAAc,mBAAY,KAAZ,C;YAGd,IAf+E,wBAejE,0EafIE,C Ae/E,C;cACI,mBAAY,gBAAZ,EAAy,wBAAZ,YAAyB,O;;cAEzB,WAAW,I;;UAGP,OAAZ,kBAAY,EAAK,IAAL,EAAW,OAAx,EAAoB,IAApB,C;;UAGE,oB;UAAuB,SAAZ,kBAAY,O;UAArC,uD;YACI,gBAAC,mBAAY,OAAZ,C;YACd,mBAAY,OAAZ,IAAqB,I;YAGrB,IA7B+E,wBA6BjE,kFA7BiE,CA6B/E,C;cACI,mBAAY,gBAAZ,EAAy,wBAAZ,YAAyB,S;;cAEzB,WAAW,I;;UAGnB,UAAU,mBAAY,OAAZ,C;UAEV,mBAAC,CAAd,YAAsB,IAAtB,Y;YACI,gBAAC,mBAAY,OAAZ,C;YACd,mBAAY,OAAZ,IAAqB,I;YAGrB,IA1C+E,wBA0CjE,kFA1CiE,CA0C/E,C;cACI,mBAAY,OAAZ,IAAuB,S;cACvB,UAAU,mBAAY,OAAZ,C;;cAEV,WAAW,I;;;QAIvB,IAAI,QAAJ,C;UACI,YAAO,mBAAY,UAAU,WAAV,IAAZ,C;QAEX,yBAAO,Q;;;MArDuD,6B;K;2CAEIE,qB;MASsB,IAII,IAJJ,EAKM,MALN,EAaA,MAbA,EAuB,MAbvB,EAKBI,MAIBJ,EAmBM,MAAnBN,EA+BI,M;MAvCb,aAAL,IAAK,U;MAAL,Y;QAA8B,SAAZ,kB3KoxOnB,YAAQ,C;;M2KpxOX,W;QACI,OOAO,K;MAEX,WA1XgD,mBAAY,cA0XnC,SA1XmC,IAAZ,C;MA2XhD,cAAc,W;MACd,eAAe,K;MAEf,IAAI,cAAO,IAAX,C;QACI,iBAAC,WAAAd,UAAyB,IAAZB,U;UACI,cAAc,mBAAY,KAAZ,C;UAGd,IAAI,UAAU,0EAAV,CAAJ,C;YACI,mBAAY,gBAAZ,EAAy,wBAAZ,YAAyB,O;;YAEzB,WAAW,I;;QAGP,OAAZ,kBAAY,EAAK,IAAL,EAAW,OAAx,EAAoB,IAApB,C;;QAGE,oB;QAAuB,SAAZ,kBAAY,O;QAArC,uD;UACI,gBAAC,mBAAY,OAAZ,C;UACd,mBAAY,OAAZ,IAAqB,I;UAGrB,IAAI,UAAU,kFAAV,CAAJ,C;YACI,mBAAY,gBAAZ,EAAy,wBAAZ,YAAyB,S;;YAEzB,WAAW,I;;QAGnB,UAAU,mBAAY,OAAZ,C;QAEV,mBAAC,CAAd,YAAsB,IAAtB,Y;UACI,gBAAC,mBAAY,OAAZ,C;UACd,mBAAY,OAAZ,IAAqB,I;UAGrB,IAAI,UAAU,kFAAV,CAAJ,C;YACI,mBAAY,OAAZ,IAAuB,S;YACvB,UAAU,mBAAY,OAAZ,C;;YAEV,WAAW,I;;;MAIvB,IAAI,QAAJ,C;QACI,YAAO,mBAAY,UAAU,WAAV,IAAZ,C;MAEX,OOAO,Q;K;iCAGX,Y;MACI,WA7agD,mBAAY,cA6anC,SA7amC,IAAZ,C;MA8ahD,IAAI,cAAO,IAAX,C;QACgB,OAAZ,kBAAY,EAAK,IAAL,EAAW,WAAx,EAAiB,IAAjB,C;;QACT,ItKpS6C,CAAC,csKoS9C,C;UACS,OAAZ,kBAAY,EAAK,IAAL,EAAW,WAAx,EAAiB,kBAAY,OAA7B,C;UACA,OAAZ,kBAAY,EAAK,IAAL,EAAW,CAAX,EAAc,IAAd,C;;MAEHb,cAAO,C;MACP,YAAO,C;K;2CAGX,iB;MAGe,IAAC,IAAD,EAeJ,M;MAfP,WACW,eAAC,OAAL,KAAM,OAAN,IAAc,SAAlB,GAAwB,KAAxB,GAAmC,aAAa,KAAb,EAAoB,SAApB,CAApC,uB;MAEX,WA7bgD,mBAAY,cA6bnC,SA7bmC,IAAZ,C;MA8bhD,IAAI,cAAO,IAAX,C;Q3J2XJ,U2J1XQ,kB3J0XR,E2J1X6B,I3J0X7B,EAD+F,CAC/F,E2J1XgD,W3J0XhD,E2J1XiE,I3J0XjE,C;;Q2JzXW,ItKpT6C,CAAC,csKoT9C,C;U3JyXX,U2JxXQ,kB3JwXR,E2JxX6B,I3JwX7B,E2JxXuD,C3JwXvD,E2JxXuE,W3JwXvE,E2JxXwF,kBAAY,O3JwXpG,C;UAAA,U2JvXQ,kB3JuXR,E2JvX6B,I3JuX7B,E2JvXuD,kBAAY,OAAZ,GAAmB,WAAAnB,I3JuXvD,E2JvX6F,C3JuX7F,E2JvX2G,I3JuX3G,C;;M2JrXI,IAAI,IAAK,OAAL,GAAy,SAAhB,C;QACI,KAAK,SAAL,IAAa,I;;MAIjB,OOAO,qD;K;mCAGX,Y;MAEI,OOAO,qBAAQ,gBAAmB,SAAnB,OAAR,C;K;+CAGX,iB;MAC0D,4BAAQ,KAAR,C;K;+CAC1D,Y;MAA0C,qB;K;IAE1C,gC;MAAA,oC;MACI,0BvHriBuC,E;MuHsiBvC,sBAaiC,U;MACjC,4BAAuC,E;K;yDAEvC,oC;MAEI,kBAAkB,eAAe,eAAGB,CAA/B,K;MACIB,IAAI,eAAc,WAAAd,QAA4B,CAAhC,C;QACI,cAAc,W;MACIB,IAAI,eAAc,UAAAd,QAA6B,CAAjC,C;QACI,cAAkB,cAAc,UAAIB,GAAgC,UAAhC,GAAmD,U;MACrE,OOAO,W;K;;IAZf,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;qDAgBA,qB;MAEI,WavegD,mBAAY,cAuenC,SAvemC,IAAZ,C;MAwehD,WAAe,kBAAa,cAAO,IAAxB,GAA8B,WAA9B,GAAwC,cAAO,kBAAY,OAAnB,I;MACnD,UAAU,IAAV,EAAGB,cAAhB,C;K;;IA5iBJ,iD;MAAA,oD;MAGwC,+B;MApB5C,sB;MAqBsB,Q;MACV,wBAAmB,CAAnB,C;QAAwB,4D;WACxB,sBAAkB,CAAlB,C;QAAuB,uBAAa,eAAb,O;;QACf,MAAM,gCAAyB,uBAAoB,eAA7C,C;MAHIB,0B;MAJJ,Y;K;IAWA,kC;MAAA,oD;MAGoB,+B;MA/BxB,sB;MAGCQ,sBAAC,qD;MAIJB,Y;K;IAOA,4C;MAAA,oD;MAG2C,+B;MATC/C,sB;MAuCQ,sBrJpB8D,YqJoBhD,QrJpBgD,C;MqJqB9D,aAAO,mBAAY,O;MACnB,IAAI,mB3KsrPD,YAA

Q,C2KtrPX,C;QAA2B,sBAAC,qD;MAN7C,Y;K;IC5BJ,4B;MAMoB,Q;M5Kq2rBA,U;MADhB,UAAe,C;MACf,u  
D;QAAgB,cAAhB,iB;QACI,YAAgB,O4Kv2rBiB,O5Ku2rBjC,I;M4Kv2rBJ,aAAa,iB5Ky2rBN,G4Kz2rBM,C;MA  
Cb,wBAAGB,SAAhB,gB;QAAgB,gBAAA,SAAhB,M;QACW,SAAP,MAAO,EAAO,SAAP,C;MAEX,OAAO,M;  
K;IAGX,0B;MASiB,Q;MAFb,YAAY,iBAAa,gBAAb,C;MACZ,YAAY,iBAAa,gBAAb,C;MACZ,wBAAa,SAAb,g  
B;QAAa,WAAA,SAAb,M;QACI,KAAM,WAAI,IAAK,MAAT,C;QACN,KAAM,WAAI,IAAK,OAAT,C;MAEV,  
OAAO,UAAAS,KAAT,C;K;gGAGX,qB;MAWW,4B;MAAA,U;QAAqB,OAAL,S5KirPhB,YAAQ,C;M4KjrPf,W;  
K;oFAGJ,mC;MAUI,O5KoqPO,qBAAQ,C4KpqPf,GAAe,cAAf,GAAMC,S;K;IAGvC,iD;MAMIL,IAAI,cAAS,KAA  
b,C;QAAoB,OAAO,I;MAC3B,IAAI,qBAAGB,aAAhB,IAAiC,SAAK,OAAL,KAAa,KAAM,OAAxD,C;QAA8D,O  
AAO,K;MAErE,4C;QACI,SAAS,UAAK,CAAL,C;QACT,SAAS,MAAM,CAAN,C;QAET,IAAI,OAAO,EAAX,C;  
UACI,Q;eACG,IAAI,cAAc,UAAIB,C;UACH,OAAO,K;QAIP,0BAAsB,kBAAtB,C;UAA4C,IAAI,CAAI,kBAAH,  
EAAG,EAakB,EAaIB,CAAR,C;YAA+B,OAAO,K;eACIF,8BAAsB,sBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EA  
G,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,+BAAsB,uBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EA  
c,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,6BAAsB,qBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAC,EA  
d,CAAR,C;YAA2B,OAAO,K;eAC9E,8BAAsB,sBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAC,EAAd,CAA  
R,C;YAA2B,OAAO,K;eAC9E,+BAAsB,uBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAC,EAAd,CAAR,C;Y  
AA2B,OAAO,K;eAC9E,gCAAsB,wBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,  
OAAO,K;eAC9E,8BAAsB,sBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,  
K;eAC9E,iCAAsB,yBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAE9  
E,qCAAsB,6BAAtB,C;UAA4C,IAAI,CAAI,gBAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,sC  
AAsB,8BAAtB,C;UAA4C,IAAI,CAAI,gBAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,oCAAs  
B,4BAAtB,C;UAA4C,IAAI,CAAI,gBAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,qCAAsB,6B  
AAtB,C;UAA4C,IAAI,CAAI,gBAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAEtE,IAAI,YAAM,EA  
N,CAAJ,C;UAAc,OAAO,K;MAIrC,OAAO,I;K;IAGX,4C;MAKI,IAAI,iBAAJ,C;QAakB,OAAO,M;MACzB,aAA  
a,CAAK,eAAL,gBAAK,EAAa,SAAb,CAAL,GAA6C,CAA7C,QAAiD,CAAjD,I;MvC6SkB,kBAAxB,mBuC5SY,  
MvC4SZ,C;MuC3SH,oDvK5BgD,gBuK4BhD,C;MADJ,O1JnCO,WmH+U6C,W;K;IuCvSxD,mE;MAEI,IAAY,SA  
AR,0BAAJ,C;QACI,MAAO,gBAAO,OAAP,C;QACP,M;MAEJ,SAAU,WAAI,SAAJ,C;MACV,MAAO,gBAAO,  
EAAP,C;MAEP,4C;QACI,IAAI,MAAK,CAAT,C;UACI,MAAO,gBAAO,IAAP,C;QAEX,cAAc,UAAK,CAAL,C;  
QAEV,IADE,OACF,S;UAAmB,MAAO,gBAAO,MAAP,C;aAC1B,mBAFE,OAef,E;UAA2B,4BAAR,OAAQ,EA  
A4B,MAA5B,EAAoC,SAAP,C;aAC3B,uBAHE,OAGF,E;UAAmB,MAAO,gBAAE,gBAAR,OAAQ,CAAF,C;aA  
C1B,wBAJE,OAIF,E;UAAmB,MAAO,gBAAE,gBAAR,OAAQ,CAAF,C;aAC1B,sBALE,OAKF,E;UAAmB,MAA  
O,gBAAE,gBAAR,OAAQ,CAAF,C;aAC1B,uBANE,OAMF,E;UAAmB,MAAO,gBAAE,gBAAR,OAAQ,CAAF,C;a  
AC1B,wBAPE,OAOF,E;UAAmB,MAAO,gBAAE,gBAAR,OAAQ,CAAF,C;aAC1B,yBARE,OAQF,E;UAAmB,M  
AAO,gBAAE,gBAAR,OAAQ,CAAF,C;aAC1B,uBATE,OASF,E;UAAmB,MAAO,gBAAE,gBAAR,OAAQ,CAAF,  
C;aAC1B,0BAVE,OAUF,E;UAAmB,MAAO,gBAAE,gBAAR,OAAQ,CAAF,C;aAE1B,kBAZE,OAYF,c;UAAmB,  
MAAO,gBAAE,kBAAR,OAAQ,CAAF,C;aAC1B,kBAbE,OAaf,e;UAAmB,MAAO,gBAAE,kBAAR,OAAQ,CAAF,  
C;aAC1B,kBADE,OAcf,a;UAAmB,MAAO,gBAAE,kBAAR,OAAQ,CAAF,C;aAC1B,kBAfE,OAef,c;UAAmB,M  
AAO,gBAAE,kBAAR,OAAQ,CAAF,C;UAEP,MAAO,gBAAO,OAAQ,WAAf,C;MAIIC,MAAO,gBAAO,EAAP,  
C;MACP,SAAU,kBAAmB,iBAAV,SAAU,CAANB,C;K;ICpJd,uC;MAIqD,+CAAwC,iBAAO,CAA/C,IAAoD,mC;  
K;IAEZg,4D;MAWQ,kBADE,SACF,O;QADJ,OACc,S;WACV,kBAFE,SAEF,c;QAEQ,yCAAwB,MAAO,KAAP,  
GAAc,CAATC,C;UAJZ,OAIuD,S;UAJvD,OAK6B,mBAAL,SAAK,CAAT,GAA+B,sBAA/B,GAAgD,S;QALpE,  
OAOgB,oCAAJ,GAA0C,sBAA1C,GAA2D,mB;K;IAG3E,gD;MAWQ,kBADE,SACF,O;QADJ,OACc,S;WACV,k  
BAFE,SAEF,c;QAFJ,OAEB8B,mBAAL,SAAK,CAAT,GAA+B,sBAA/B,GAAgD,S;QAFrE,OAGgB,oCAAJ,GAA  
0C,sBAA1C,GAA2D,mB;K;IAG3E,kD;MAKI,OAAL,oCAAJ,GAA0C,sBAA1C,GAA2D,oB;K;IAE/D,kD;MAKI,  
OAAL,oCAAJ,GAA0C,oBAA1C,GAA2D,iB;K;IxKnD/D,yB;MAAA,6B;K;sCACI,Y;MAAkC,Y;K;0CACIC,Y;M  
AAsC,Y;K;wCACtC,Y;MAAgC,Q;K;4CACHC,Y;MAAoC,S;K;mCACpC,Y;MAA+B,MAAM,6B;K;uCACrC,Y;M  
AAmC,MAAM,6B;K;IAN7C,qC;MAAA,oC;QAAA,mB;MAAA,6B;K;IASA,qB;MAAA,yB;MACI,+C;K;iCAE  
A,iB;MAA4C,qCAoB,KAAM,U;K;mCACtE,Y;MAA+B,Q;K;mCAC/B,Y;MAAkC,W;K;iFAEX,Y;MAAQ,Q;K;  
kCAC/B,Y;MAAkC,W;K;yCACIC,mB;MAAmD,Y;K;8CACnD,oB;MAAmE,OAAA,QAAS,U;K;sCAE5E,iB;MA

AwC,MAAM,8BAA0B,iDAA8C,KAA9C,MAA1B,C;K;wCAC9C,mB;MAA8C,S;K;4CAC9C,mB;MAAkD,S;K;m  
CAEID,Y;MAA6C,kC;K;uCAC7C,Y;MAAqD,kC;K;+CACrD,iB;MACI,IAAI,UAAS,CAAb,C;QAAgB,MAAM,8  
BAA0B,YAAS,KAAnc,C;MACtB,OAAO,2B;K;0CAGX,8B;MACI,IAAI,cAAa,CAAb,IAAkB,YAAW,CAAjC,C;  
QAAoC,OAAO,I;MAC3C,MAAM,8BAA0B,gBAaA,SAAb,mBAaKc,OAA5D,C;K;wCAGV,Y;MAAiC,8B;K;;I  
A5BrC,iC;MAAA,gC;QAAA,e;;MAAA,yB;K;IA+BA,iC;MAA8D,6BAaKb,SAaIB,EAAoC,KAApC,C;K;IAE5B,  
8C;MAAC,oB;MAA0B,0B;K;yFACIC,Y;MAAQ,OAAA,WAAO,O;K;0CACtC,Y;MAAkC,OAAA,WL4qP3B,YA  
AQ,C;K;iDK3qPf,mB;MAA6C,OAAO,SAAP,WAAO,EAAS,OAAT,C;K;sDACpD,oB;MAAsE,c;;QgB8nDtD,Q;Q  
ADhB,IAAI,chB7nDyD,QgB6nDzD,iBhB7nDyD,QgB6nDnC,UAA1B,C;UAAqC,aAAO,I;UAAp,e;;QACrB,OhB9  
nD6C,QgB8nD7C,W;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,ChB9nDkD,oBgB8nDvC,OhB9nDuC,  
CgB8nDtD,C;YAAyB,aAAO,K;YAAP,e;;;QAC/C,aAAO,I;;MhB/nDsD,iB;K;2CAC7D,Y;MAAuC,OAAO,qBAA  
P,WAAO,C;K;0CAC9C,Y;MAC+C,gBAAP,W;MAAA,OAAwB,cAAxB,GiBiKpC,SjBjKoC,GiBmKpC,SN63BoB  
,Q;K;;IX7hC5B,qB;MAIsC,8B;K;IAEtC,4B;MAIqD,OAAI,QAAS,OAAT,GAAgB,CAApB,GAAgC,OAAT,QAAS  
,CAAhC,GAA8C,W;K;mFAEnG,yB;MAAA,qD;MAAA,mB;QAK0C,kB;O;KAL1C,C;+FAOA,yB;MAAA,+D;M  
AAA,mB;QAMwD,uB;O;KANxD,C;2FAQA,yB;MAAA,+D;MAAA,mB;QAMoD,uB;O;KANpD,C;IAQA,mC;M  
AKI,OAAI,QAAS,OAAT,KAAiB,CAArB,GAAwB,gBAaXB,GAAyC,iBAAU,sBAaKB,QAAIB,EAAwC,IAAxC,  
CAAV,C;K;IAE7C,iC;MAKI,OAAI,QAAS,OAAT,KAAiB,CAArB,GAAwB,gBAaXB,GAAyC,iBAAU,sBAaKB,  
QAAIB,EAAwC,IAAxC,CAAV,C;K;IAE7C,gC;MAI2D,OAAI,eAAJ,GAAqB,OAAO,OAAP,CAArB,GAA0C,W;  
K;IAErG,mC;MAImE,OAAS,cAAT,QAAS,C;K;gFAE5E,yB;MAaA,gE;MAbA,6B;QAYBI,WAAW,eAduE,IAcvE,  
C;QaCX,iBAaC,CAAd,UbfkF,IaeIF,U;UbA6B,eAf2D,IAevD,CaCtB,KbDsB,CAAJ,C;;QAFyC,OAGB/D,I;O;KA3B  
X,C;8FAaA,yB;MAAA,gE;MAAA,6B;QAYI,WAAW,eAAa,IAAb,C;QaCX,iBAaC,CAAd,UbAO,IaAP,U;UbA6B,  
eAAI,KaCtB,KbDsB,CAAJ,C;;QAC7B,OAAO,I;O;KAdX,C;wFAiBA,yB;MiBzFA,+D;MjByFA,gC;QiBrF0B,gBA  
Af,gB;QjBsGkB,aa5FzB,W;Qb4FA,Oa3FO,SIXoC,Q;O;KjBqF/C,C;yFAyBA,yB;MiB3GA,4E;MAAA,gE;MjB2G  
A,0C;QiBvGI,qBjB4HyB,QiB5HzB,C;QAC8B,gBAAvB,ejB2HkB,QiB3HIB,C;QjB2H4B,aazHnC,W;QbyHA,Oax  
HO,SIH4C,Q;O;KjBsGvD,C;IAkCI,mC;MAAQ,uBAAG,iBAAO,CAAP,IAAH,C;K;IAQR,qC;MAAQ,OAAA,SA  
AK,KAAL,GAAY,CAAZ,I;K;4FAEZ,qB;MAK4D,QAAC,mB;K;kGAE7D,qB;MAWI,OAAO,qBAAGB,SAAK,U;  
K;sFAGhC,yB;MAAA,qD;MAAA,4B;QAKgE,uCAAQ,W;O;KALxE,C;sFAOA,yB;MAAA,qD;MAAA,4B;QAKo  
D,uCAAQ,W;O;KAL5D,C;sFAOA,mC;MASI,OAAI,mBAAJ,GAAe,cAAf,GAAmC,S;K;4FAGvC,+B;MAQoH,O  
AAA,SAAK,qBAAY,QAAC,C;K;IAGzH,uC;MAK+E,kBAAhB,0B;MAAwB,+B;MAAxB,Oa9MpD,W;K;IbiNX,y  
C;MAAkD,QAAM,cAAN,C;aAC9C,C;UAD8C,OACzC,W;aACL,C;UAF8C,OAEzC,OAAO,sBAAK,CAAL,CAA  
P,C;;UAFyC,OAGtC,S;;K;IAGZ,8D;MAGbKE,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MACjG,WAAW,  
cAAX,EAAiB,SAAjB,EAA4B,OAA5B,C;MAEA,UAAU,S;MACV,WAAW,UAAU,CAAV,I;MAEX,OAAO,OAA  
O,IAAd,C;QACI,UAAW,GAAY,GAAN,IAAM,KAAK,C;QAC5B,aAAa,sBAAI,GAAJ,C;QACb,UAAU,cAAc,M  
AAAd,EAAsB,OAAtB,C;QAEV,IAAI,MAAM,CAAV,C;UACI,MAAM,MAAM,CAAN,I;aACL,IAAI,MAAM,CAA  
V,C;UACD,OAAO,MAAM,CAAN,I;UAEP,OAAO,G;;MAEf,OAAO,EAAE,MAAM,CAAN,IAAF,K;K;IAGX,4E  
;MAe8E,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MAC7G,WAAW,cAAX,EAAiB,SAAjB,EAA4B,OAA  
5B,C;MAEA,UAAU,S;MACV,WAAW,UAAU,CAAV,I;MAEX,OAAO,OAAO,IAAd,C;QACI,UAAW,GAAY,GA  
AN,IAAM,KAAK,C;QAC5B,aAAa,sBAAI,GAAJ,C;QACb,UAAU,UAAW,SAAQ,MAAR,EAAgB,OAAhB,C;QA  
ErB,IAAI,MAAM,CAAV,C;UACI,MAAM,MAAM,CAAN,I;aACL,IAAI,MAAM,CAAV,C;UACD,OAAO,MAA  
M,CAAN,I;UAEP,OAAO,G;;MAEf,OAAO,EAAE,MAAM,CAAN,IAAF,K;K;kGAGX,yB;MAAA,8D;MAAA,4  
D;MAsBqC,8D;QAAA,qB;UAAE,qBAAc,iBAAS,EAAT,CAAd,EAA4B,WAA5B,C;S;O;MAtBvC,+D;QAKBI,yB;  
UAAA,YAAiB,C;QACjB,uB;UAAA,UAAe,c;QAGf,+BAaA,SAAb,EAAwB,OAAxB,EAAiC,oCAAjC,C;O;KAtB  
J,C;IA6BA,mE;MAmBoC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MACnE,WAAW,cAAX,EAAiB,SAA  
jB,EAA4B,OAA5B,C;MAEA,UAAU,S;MACV,WAAW,UAAU,CAAV,I;MAEX,OAAO,OAAO,IAAd,C;QACI,U  
AAW,GAAY,GAAN,IAAM,KAAK,C;QAC5B,aAAa,sBAAI,GAAJ,C;QACb,UAAU,WAAW,MAAX,C;QAEV,IA  
AI,MAAM,CAAV,C;UACI,MAAM,MAAM,CAAN,I;aACL,IAAI,MAAM,CAAV,C;UACD,OAAO,MAAM,CAA  
N,I;UAEP,OAAO,G;;MAEf,OAAO,EAAE,MAAM,CAAN,IAAF,K;K;IAGX,8C;MAMQ,gBAAY,OAAZ,C;QAA  
uB,MAAM,gCAAYB,gBAaA,SAAb,mCAAKD,OAAID,OAAzB,C;WAC7B,gBAAY,CAAZ,C;QAAiB,MAAM,8B  
AA0B,gBAaA,SAAb,yBAA1B,C;WACvB,cAAU,IAAV,C;QAAkB,MAAM,8BAA0B,cAAW,OAAAX,gCAA2C,IA



A3C,OAA1B,C;K;IAChC,8B;MAEoC,MAAM,wBAAoB,8BAApB,C;K;IAE1C,8B;MAEoC,MAAM,wBAAoB,8B  
AApB,C;K;;;wFyGnb1C,yB;MxGgCA,wE;MwGhCA,uC;QAmBW,kBxGqBiD,oB;QwGM9C,Q;QAAA,OAAK,0  
B;QAaf,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBAaKB,sBAAY,GAAZ,C;UAC1  
B,WxGyKJ,awGzKgB,GxGyKhB,EwGvMyC,SA8BIB,CAAU,GAAV,EAAe,WAAf,EAA4B,CAA5B,EAA+B,uB  
AAuB,CAAC,WAAy,mBAAY,GAAZ,CAAnE,CxGyKvB,C;;QwGvMA,OAgCO,W;O;KAnDX,C;4FAsBA,6C;M  
AwBc,Q;MAAA,OAAA,SAAK,iB;MAAf,OAAU,cAAV,C;QAAU,mB;QACN,UAAU,sBAAM,CAAN,C;QACV,k  
BAaKB,sBAAY,GAAZ,C;QAC1B,WxGyKJ,awGzKgB,GxGyKhB,EwGzKuB,UAAU,GAAV,EAAe,WAAf,EAA4  
B,CAA5B,EAA+B,uBAAuB,CAAC,WAAy,mBAAY,GAAZ,CAAnE,CxGyKvB,C;;MwGvKA,OAAO,W;K;iFAG  
X,yB;MAAA,gB;MAAA,8B;MxGtBA,wE;MwGsBA,6D;QAnCW,kBxGqBiD,oB;QwGM9C,Q;QAAA,OAAK,0B;  
QAaf,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBAaKB,sBAAY,GAAZ,C;UA8Bw  
E,U;UA7B1F,WxGyKJ,awGzKgB,GxGyKhB,EwG5IkC,UA7BD,GA6BC,EA7BoB,uBAAuB,CAAC,WAAy,mBA  
AY,GAAZ,CA6BzC,GAAW,qBA7B3B,GA6B2B,EA7BT,CA6BS,CAAX,GAA6C,UA7BxD,WA6BwD,6DAA5D,  
EA7BiB,CA6BjB,CxG4IIC,C;;QwG7IA,OA1BO,W;O;KAGX,C;kFA0BA,yB;MAAA,gB;MAAA,8B;MAAA,0E;Q  
AlCc,Q;QAAA,OAAK,0B;QAaf,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBA6DQ,  
WA7DU,WAAy,GAAZ,C;UA6DuF,U;UAAjG,WxG6GZ,awGzKgB,GxGyKhB,EwG7GiD,UA5DhB,GA4DgB,E  
A5DK,uBAAuB,CA4DjE,WA5D8E,mBAAY,GAAZ,CA4D1B,GAAW,qBA5D1C,GA4D0C,EA5DxB,CA4DwB,C  
AAX,GAA6C,UA5DvE,WA4DuE,6DAA5D,EA5DE,CA4DF,CxG6GjD,C;;QwG9GA,OACY,W;O;KA7BhB,C;iF  
AgCA,yB;MAAA,gB;MAAA,8B;MxGhFA,wE;MwGgFA,qD;QA7FW,kBxGqBiD,oB;QwGM9C,Q;QAAA,OAA  
K,0B;QAaf,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBAaKB,sBAAY,GAAZ,C;U  
AkFiD,U;UAjFnE,WxGyKJ,awGzKgB,GxGyKhB,EwGxFgC,UAjFsB,uBAAuB,CAAC,WAAy,mBAAY,GAAZ,  
CAiFhD,kBAA6B,UAjFjC,WaiFiC,6DAAvC,EAjFmB,CAiFnB,CxGwFhC,C;;QwGzFA,OA9EO,W;O;KA6DX,C;  
oFAoBA,yB;MAAA,gB;MAAA,8B;MAAA,kE;QAtFc,Q;QAAA,OAAK,0B;QAaf,OAAU,cAAV,C;UAAU,mB;U  
ACN,UAAU,sBAAM,CAAN,C;UACV,kBA2GQ,WA3GU,WAAy,GAAZ,C;UA2GgE,U;UAA1E,WxG+DZ,awGz  
KgB,GxGyKhB,EwG/D+C,UA1GO,uBAAuB,CA0GjE,WA1G8E,mBAAY,GAAZ,CA0GjC,kBAA6B,UA1GhD,W  
A0GgD,6DAAvC,EA1GI,CA0GJ,CxG+D/C,C;;QwGhEA,OACY,W;O;KAvBhB,C;qFA0BA,yB;MAAA,gB;MAA  
A,8B;MxG9HA,wE;MwG8HA,uC;QA3IW,kBxGqBiD,oB;QwGM9C,Q;QAAA,OAAK,0B;QAaf,OAAU,cAAV,C  
;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBAaKB,sBAAY,GAAZ,C;UACC,oB;UakIc,U;UAAjC,IAI  
IkD,uBAAuB,CAAC,WAAy,mBAAY,GAAZ,CAkItF,C;YADA,mBAjI+C,C;;YAiI/C,mBACKB,UAIIW,GAkIX,E  
AAe,UAIIC,WakID,6DAAf,EAI6B,CAkI7B,C;;UAIIB,WxGyKJ,awGzKgB,GxGyKhB,mB;;QwGzCA,OA9HO,  
W;O;KA2GX,C;sFAwBA,yB;MAAA,gB;MAAA,8B;MAAA,oD;QAxIc,Q;QAAA,OAAK,0B;QAaf,OAAU,cAAV  
,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBA6JQ,WA7JU,WAAy,GAAZ,C;UACC,oB;UA8Jc,U;  
UAAjC,IA9JkD,uBAAuB,CA4JjE,WA5J8E,mBAAY,GAAZ,CA8JtF,C;YADA,mBA7J+C,C;;YA6J/C,mBACKB,U  
A9JW,GA8JX,EAAe,UA9JC,WA8JD,6DAAf,EA9J6B,CA8J7B,C;;UAFV,WxGaZ,awGzKgB,GxGyKhB,mB;;Qw  
GbA,OAAy,W;O;KAvBhB,C;IA6BA,6C;MArKc,Q;MAAA,OAAK,0B;MAAf,OAAU,cAAV,C;QAAU,mB;QAC  
N,UAAU,sBAAM,CAAN,C;QACV,kBA+KG,WA/Ke,WAAy,GAAZ,C;QA2GgE,U;QAoE/E,WxGLP,awGzKgB,  
GxGyKhB,EwGKmC,CA9KmB,uBAAuB,CA8KtE,WA9KmF,mBAAY,GAAZ,CA0GjC,GAoErC,CAPeQc,GAA6  
B,UA1GhD,WA0GgD,6DAoEnD,IAAM,CAAN,IxGLnC,C;;MwGKA,OAAO,W;K;IgeNp0B,oC;MAAC,kB;MAA  
uB,kB;K;;wCAN7D,Y;MAMsC,iB;K;wCANtC,Y;MAM6D,iB;K;0CAN7D,wB;MAAA,wBAMsC,qCANtC,EAM6  
D,qCAN7D,C;K;sCAAA,Y;MAAA,OAMsC,mDANtC,IAM6D,wCAN7D,O;K;sCAAA,Y;MAAA,c;MAMsC,sD;  
MAAuB,sD;MAN7D,a;K;oCAAA,iB;MAAA,4IAMsC,sCANtC,IAM6D,sCAN7D,I;K;wFjKEA,yB;MAAA,kC;M  
AAA,4C;MAAA,kD;QAMuF,wC;O;MANvF,4CAOI,Y;QAAuC,8B;O;MAP3C,8E;MAAA,2B;QAMuF,2C;O;KA  
NvF,C;IAcsC,2C;MAAC,wC;K;0CACnC,Y;MAAqD,4BAAiB,wBAAjB,C;K;;IAIzD,yC;MAI4D,OAAI,oCAAJ,G  
AA2B,SAAK,KAhC,GAA0C,I;K;IAEtG,uD;MAI0E,OAAI,oCAAJ,GAA2B,SAAK,KAhC,GAA0C,S;K;IAGp  
H,8B;MAMoB,Q;MADhB,aAAa,gB;MACG,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAQL,OAAP,MAAO,E  
AAO,OAAP,C;;MAEX,OAAO,M;K;IAGX,4B;MAUiB,Q;MAHb,mBAAmB,mCAAwB,EAAxB,C;MACnB,YAA  
Y,iBAAa,YAAb,C;MACZ,YAAy,iBAAa,YAAb,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,KAAM,W  
AAI,IAAK,MAAT,C;QACN,KAAM,WAAI,IAAK,OAAT,C;;MAEV,OAAO,UAAS,KAAT,C;K;wFUxDX,qB;MA  
KqE,gB;K;IAErE,iC;MAMoE,4BAAiB,SAAJB,C;K;uFAEpE,gC;MAKI,OAAGB,mBAAhB,C;QAAGB,8B;QAAM,

UAAU,OAAV,C;;K;IAMY,oC;MAAC,0B;MACnC,eAAoB,C;K;yCACpB,Y;MAAwC,OAAA,eAAS,U;K;sCACjD ,Y;MAA6E,Q;MAAhC,wBAAa,oBAAmB,mBAAnB,EAAMb,2BAAnB,QAAb,EAA0C,eAAS,OAAAnD,C;K;;sFwJ 5BjD,yB;MAAA,4E;MAAA,gB;MAAA,8B;MAAA,+C;QAUiC,Q;QAA7B,OAA6B,wCAAqB,QAAS,aAA9B,0D; O;KAVjC,C;wFAYA,yB;MAAA,4E;MAAA,gB;MAAA,8B;MAAA,+C;QAWiC,Q;QAA7B,OAA6B,wCAAqB,Q AAS,aAA9B,0D;O;KAXjC,C;sFAaA,+C;MAQI,SAAK,aAAI,QAAS,aAAb,EAAMb,KAAAnB,C;K;ICnCT,8C;MA UI,IAAI,wCAAJ,C;QACI,OAAO,SAAK,4BAAqB,GAARb,C;MAET,4B;M1KsTI,Q;MALX,YAAY,oB0KjTa,G1K iTb,C;MACZ,IAAI,iBAAiB,CAAC,4B0KITG,G1KkTH,CAAtB,C;Q0KITgC,MAAM,2BAAuB,wCAAvB,C;;Q1Ks TIC,2BAAO,sE;;M0KtTX,+B;K;IAGJ,8C;MAUQ,kBADE,SACF,kB;QADJ,OACkC,YAAT,SAAK,IAAI,EAAY,Y AAZ,C;;QADIC,OAeY,uBAAMb,SAAnB,EAAY,YAAzB,C;K;IAGhB,gD;MAWQ,kBADE,SACF,yB;QADJ,OA CyC,cAAT,SAAK,IAAI,EAAY,YAAZ,C;;QADzC,OAeY,8BAA0B,SAA1B,EAAGC,YAAhC,C;K;IAC0B,4C; MAAC,wB;MAAoC,0B;K;qEAPc,Y;MAAA,yB;K;0CACvC,iB;MAA4C,OAAI,OAAJ,QAAl,EAAO,KAAP,C;K ;4CACHd,Y;MAA+B,OAAI,SAAJ,QAAl,C;K;4CACnC,Y;MAAkC,OAAA,QAAl,W;K;0FACf,Y;MAAQ,OAAA, QAAl,K;K;2CACnC,Y;MAAkC,OAAA,QAAl,U;K;qDACtC,e;MAA4C,OAAA,QAAl,mBAAY,GAAZ,C;K;uDA ChD,iB;MAAGe,OAAA,QAAl,qBAAc,KAAd,C;K;6CACpE,e;MAA+B,OAAA,QAAl,WAAI,GAAJ,C;K;0FACT, Y;MAAQ,OAAA,QAAl,K;K;4FACH,Y;MAAQ,OAAA,QAAl,O;K;6FACJ,Y;MAAQ,OAAA,QAAl,Q;K;8DAEv D,e;MAAMd,gBAAJ,Q;MAAI,4B;M1K8PxC,Q;MALX,YAAY,oB0KzPyD,G1KyPzD,C;MACZ,IAAI,iBAAiB,C AAC,4B0K1P+C,G1K0P/C,CAAtB,C;QACI,2B0K3PwE,mB;;Q1K8PxE,2BAAO,sE;;M0K9PoC,+B;K;;IAGN,mD ;MAAC,wB;MAA2C,0B;K;4EAA3C,Y;MAAA,yB;K;iDAC1C,iB;MAA4C,OAAI,OAAJ,QAAl,EAAO,KAAP,C; K;mDACHd,Y;MAA+B,OAAI,SAAJ,QAAl,C;K;mDACnC,Y;MAAkC,OAAA,QAAl,W;K;iGACf,Y;MAAQ,OA AA,QAAl,K;K;kDACnC,Y;MAAkC,OAAA,QAAl,U;K;4DACtC,e;MAA4C,OAAA,QAAl,mBAAY,GAAZ,C;K;8 DACHd,iB;MAAGe,OAAA,QAAl,qBAAc,KAAd,C;K;oDACpE,e;MAA+B,OAAA,QAAl,WAAI,GAAJ,C;K;iGA CF,Y;MAAQ,OAAA,QAAl,K;K;mGACH,Y;MAAQ,OAAA,QAAl,O;K;oGACU,Y;MAAQ,OAAA,QAAl,Q;K;sD AE5E,sB;MAAYc,OAAA,QAAl,aAAI,GAAJ,EAAS,KAAT,C;K;uDAC7C,e;MAAkC,OAAA,QAAl,cAAO,GAAP ,C;K;yDACtC,gB;MAA2C,QAAl,gBAAO,IAAP,C;K;gDAC/C,Y;MAAuB,QAAl,Q;K;qEAE3B,e;MAAMd,gBAA J,Q;MAAI,4B;M1KyOxC,Q;MALX,YAAY,oB0KpOyD,G1KoOzD,C;MACZ,IAAI,iBAAiB,CAAC,4B0KrO+C,G 1KqO/C,CAAtB,C;QACI,2B0KtOwE,mB;;Q1KyOxE,2BAAO,sE;;M0KzOoC,+B;K;;I1KvFnD,oB;MAAA,wB;M ACI,8C;K;gCAEA,iB;MAA4C,oCAAsB,KAAM,U;K;kCACxY,Y;MAA+B,Q;K;kCAC/B,Y;MAAkC,W;K;gFAEX ,Y;MAAQ,Q;K;iCAC/B,Y;MAAkC,W;K;2CAEIC,e;MAA+C,Y;K;6CAC/C,iB;MAAsD,Y;K;mCACtD,e;MAAwC, W;K;mFACY,Y;MAAQ,6B;K;gFAC/B,Y;MAAQ,6B;K;kFACI,Y;MAAQ,8B;K;uCAEjD,Y;MAAiC,6B;K;;IAjBr C,gC;MAAA,+B;QAAA,c;;MAAA,wB;K;IAoBA,oB;MAMuE,Q;MAA7B,OAA6B,uE;K;IAEvE,wB;MAAl,OAAI, KAAM,OAAAn,GAAa,CAAjB,GAA0B,QAAN,KAAM,EAAM,qBAAc,YAAY,KAAM,OAAIB,CAAd,CAAN,CA A1B,GAA6E,U;K;kFAEjF,yB;MAAA,oD;MAAA,mB;QAO8C,iB;O;KAP9C,C;8FASA,yB;MAAA,wE;MAAA,m B;QAQ4D,2B;O;KAR5D,C;IAUA,+B;MAyID,gBAA7C,qBAAoB,YAAY,KAAM,OAAIB,CAApB,C;MAAqD,w B;MAArD,OYJO,S;K;wFZMX,yB;MAAA,4D;MAAA,mB;QAOsD,qB;O;KAPtD,C;IASA,4B;MAM8G,gBAAvC, eAAc,YAAY,KAAM,OAAIB,CAAd,C;MAA+C,wB;MAA/C,OYrB5D,S;K;4FZuBX,yB;MAAA,wE;MAAA,mB; QAK8D,2B;O;KAL9D,C;IAOA,8B;MAU+E,OAAM,QAAN,KAAM,EAAM,qBAAc,YAAY,KAAM,OAAIB,CAA d,CAAN,C;K;sFAErF,yB;MgBfA,wE;MhBeA,gC;QgBXiC,gBAAtB,oB;QhB8BiB,aY9DxB,W;QZ8DA,OY7DO,S I+B2C,Q;O;KhBWtD,C;uFA2BA,yB;MgBnCA,uE;MhBmCA,0C;QgB/ByC,gBAA9B,mBhBsDiB,QgBtDjB,C;Qh BsD2B,aY7FIC,W;QZ6FA,OY5FO,SIsCmD,Q;O;KhB+B9D,C;4FAqCA,qB;MAK+D,QAAC,mB;K;kGAhEh,qB; MAWI,OAAO,qBAAGb,mB;K;sFAG3B,yB;MAAA,oD;MAAA,4B;QAM2D,uCAAQ,U;O;KANnE,C;sFAQA,mC; MASI,OAAI,mBAAJ,GA Ae,cAAf,GAAMc,S;K;yFAEvC,yB;MAyBA,kC;MAAA,8B;MAzBA,iC;QAGCiC,Q;QA xB2E,OAwBxD,CAAnB,wDAAMb,oBAxBoE,GAwBpE,C;O;KAhCpD,C;+EAUA,yB;MAAA,kC;MAAA,8B;MA AA,iC;QAKiC,Q;QAA7B,OAAgD,CAAnB,wDAAMb,YAAI,GAAJ,C;O;KALpD,C;+EAOA,iC;MAKI,sBAAl,G AAJ,EAAS,KAAT,C;K;4FAGJ,yB;MAAA,kC;MAAA,8B;MAAA,iC;QAOiC,Q;QAA7B,OAAgD,CAAnB,wDAA mB,oBAAY,GAAZ,C;O;KAPpD,C;gGASA,4B;MASsG,OAAA,SAAK,qBAAc,KAAd,C;K;kFAG3G,yB;MAAA,g D;MAAA,8B;MAAA,iC;QASiC,Q;QAA7B,OAAuD,CAA1B,+DAA0B,eAAO,GAAP,C;O;KAT3D,C;6FAWA,qB; MAWoE,oB;K;6FAEpE,qB;MAWoE,sB;K;kFAEpE,yB;MAAA,6B;MAAA,4B;QAIgE,qBAAK,aAAL,EAAU,eA AV,C;O;KAJhE,C;2FAMA,wC;MAMiF,Q;MAAA,mCAAI,GAAJ,oBAAY,c;K;uGAG7F,yB;MAAA,gB;MAAA,8

B;MAAA,+C;QAMe,Q;QALX,YAAy,oBAAI,GA AJ,C;QACZ,IAAI,iBAAiB,CAAC,4BAAY,GA AZ,CAAtB,C;U  
ACI,OAAO,c;;UAGP,OAAO,sE;;O;KANf,C;IAUA,oC;MAUkD,uCAAqB,GAArB,C;K;sFAEID,wC;MAUW,Q;M  
ADP,YAAy,oBAAI,GA AJ,C;MACL,IAAI,aAAJ,C;QACH,aAAa,c;QACb,sBAAI,GA AJ,EAAS,MAAT,C;QACA,  
a;;QAEA,Y;;MALJ,W;K;wFASJ,qB;MAMwF,OAAA,iBAAQ,W;K;wFAEHg,qB;MAMgH,OAAA,iBAAQ,W;K;4  
FAExH,6C;Meq1BoB,Q;MAAA,Ofh1BT,iBeg1BS,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;Qfh1Ba,Wei1Bb,a  
AAgB,Ofj1Be,Iei1B/B,Efj1BsC,Sei1BZ,CAAe,OAAf,CAA1B,C;;Mfj1BhB,OAA6B,W;K;wFAGjC,6C;Me60BoB,  
Q;MAAA,Ofr0BT,iBeq0BS,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;Qfr0Ba,Wes0Bb,aft0B0B,Ses0BtB,CAAY  
,OAAZ,CAAJ,EAAYC,Of0BC,Mes0B1C,C;;Mft0BhB,OAA6B,W;K;IAGjC,kC;MAIyB,Q;MAArB,wBAAqB,KA  
ArB,gB;QAAqB,aAAA,KAArB,M;QAAK,IAAC,yBAAD,EAAM,2B;QACP,sBAAI,GA AJ,EAAS,KAAT,C;;K;IA  
IR,oC;MAIyB,Q;MAAA,uB;MAArB,OAAqB,cAArB,C;QAAqB,wB;QAAhB,IAAC,yBAAD,EAAM,2B;QACP,sB  
AAI,GA AJ,EAAS,KAAT,C;;K;IAIR,oC;MAIyB,Q;MAAA,uB;MAArB,OAAqB,cAArB,C;QAAqB,wB;QAAhB,I  
AAC,yBAAD,EAAM,2B;QACP,sBAAI,GA AJ,EAAS,KAAT,C;;K;wFAIR,yB;MAAA,0D;MAAA,uE;MAAA,uC;  
QASW,kBAAY,mBAAoB,YAAy,cAAZ,CAApB,C;Qe8xBH,Q;QAAA,Ofh1BT,iBeg1BS,W;QAAhB,OAAgB,cA  
AhB,C;UAAgB,yB;Ufh1Ba,Wei1Bb,aAAgB,Ofj1Be,Iei1B/B,Ef/xB2C,Se+xBjB,CAAe,OAAf,CAA1B,C;;Qf/xBhB  
,OAI6B,W;O;KAYCjC,C;oFAYA,yB;MAAA,0D;MAAA,uE;MAAA,uC;QAYW,kBAAU,mBAAoB,YAAy,cAA  
Z,CAApB,C;Qe+wBD,Q;QAAA,Ofr0BT,iBeq0BS,W;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;Ufr0Ba,Wes0Bb,afh  
xByC,SegxBrc,CAAY,OAAZ,CAAJ,EAAYC,Of0BC,Mes0B1C,C;;QfhxBhB,OAtD6B,W;O;KA0CjC,C;0FAeA,y  
B;MAAA,wE;MAAA,uC;QAQkB,Q;QADd,aAAa,oB;QACC,OAAA,SA3FsE,QAAQ,W;QA2F5F,OAAc,cAAAd,C;  
UAAc,uB;UACV,IAAI,UAAU,KAAM,IAAhB,CAAJ,C;YACI,MAAO,aAAI,KAAM,IAAV,EAAe,KAAM,MAAr  
B,C;;QAGf,OAAO,M;O;KAbX,C;8FAGBA,yB;MAAA,wE;MAAA,uC;QAQkB,Q;QADd,aAAa,oB;QACC,OAA  
A,SA3GsE,QAAQ,W;QA2G5F,OAAc,cAAAd,C;UAAc,uB;UACV,IAAI,UAAU,KAAM,MAAhB,CAAJ,C;YACI,M  
AAO,aAAI,KAAM,IAAV,EAAe,KAAM,MAArB,C;;QAGf,OAAO,M;O;KAbX,C;yFAiBA,6C;MAOoB,Q;MAA  
A,OAAA,SA3HoE,QAAQ,W;MA2H5F,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UAC  
I,WAAy,aAAI,OAAQ,IAAZ,EAAiB,OAAQ,MAAzB,C;;MAGpB,OAAO,W;K;qFAGX,yB;MAAA,wE;MAAA,u  
C;QAOW,kBAAS,oB;QafA,Q;QAAA,OA3HoE,iBAAQ,W;QA2H5F,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAcm  
C,SAd/B,CAAU,OAAV,CAAJ,C;YACI,WAAy,aAAI,OAAQ,IAAZ,EAAiB,OAAQ,MAAzB,C;;QAapB,OAVO,  
W;O;KAGX,C;+FAUA,6C;MAOoB,Q;MAAA,OAAA,SAPJoE,QAAQ,W;MAoJ5F,OAAgB,cAAhB,C;QAAgB,yB  
;QACZ,IAAI,CAAC,UAAU,OAAV,CAAL,C;UACI,WAAy,aAAI,OAAQ,IAAZ,EAAiB,OAAQ,MAAzB,C;;MA  
GpB,OAAO,W;K;2FAGX,yB;MAAA,wE;MAAA,uC;QAOW,kBAAY,oB;QafH,Q;QAAA,OAPJoE,iBAAQ,W;Q  
AoJ5F,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,CACK,SADjC,CAAU,OAAV,CAAL,C;YACI,WAAy,aAAI,O  
AAQ,IAAZ,EAAiB,OAAQ,MAAzB,C;;QAapB,OAVO,W;O;KAGX,C;IAUA,0B;MAQqB,IAAN,I;MADX,IAAI,  
oCAAJ,C;QACW,QAAM,cAAN,C;eACH,C;YAAK,iB;YAAL,K;eACA,C;YAAK,aAAU,8BAAJ,GAakB,sBAAK  
,CAAL,CAAIb,GAA+B,oBAAW,OAAhD,C;YAAL,K;;YACQ,0BAAM,qBAAoB,YAAy,cAAZ,CAApB,CAAN,  
C;Yahl,K;;QAAP,W;;MAMJ,OAAoC,oBAA7B,mBAAM,oBAAN,CAA6B,C;K;IAGx,C,yC;MAIwB,SAAPB,W  
AAoB,Y;MAAPB,kB;K;IAEJ,4B;MAM6D,QAAM,gBAAN,C;aAcZD,C;UADyD,OACpD,U;aACL,C;UAFyD,OA  
EpD,MAAM,UAak,CAAL,CAAN,C;;UAFoD,OAGjD,mBAAM,qBAAoB,YAAy,gBAAZ,CAApB,CAAN,C;;K;I  
AGZ,yC;MAIwB,OAApB,WAAoB,Y;MAAPB,kB;K;IAEJ,4B;MAM4D,OAA6B,oBAA7B,mBAAM,oBAAN,CA  
A6B,C;K;IAEzF,yC;MAIwB,SAAPB,WAAoB,Y;MAAPB,kB;K;IAEJ,4B;MAMqD,QAAM,cAAN,C;aCjD,C;UA  
DiD,OAC5C,U;aACL,C;UAFiD,OgBhY8B,uB;;UhBgY9B,OAGzC,uB;;K;IAGZ,iC;MAMmE,4BAAC,SAAd,C;K;I  
AEnE,yC;MAKI,WAAoB,0B;MAAPB,kB;K;IAEJ,kC;MAOI,Q;MAAA,IAAI,SAAK,UAAT,C;QAAA,OAAoB,M  
AAM,IAAN,C;;QAAqC,kBAApB,qBAAC,SAAd,C;QAA4B,wBAAS,UAAT,EAAqB,WAArB,C;QAAjE,OYliBO,  
W;;MZkiBP,W;K;IAEJ,mC;MAOI,Q;MAAA,IAAI,SAAK,UAAT,C;QAAA,OAA0B,MAAN,KAAM,C;;QAAiC,k  
BAApB,qBAAC,SAAd,C;QAA4B,4B;QAAne,OY3iBO,W;;MZ2iBP,W;K;IAEJ,mC;MAOI,Q;MAAA,IAAI,SAAK  
,UAAT,C;QAAA,OAA0B,QAAN,KAAM,C;;QAAiC,kBAApB,qBAAC,SAAd,C;QAA4B,0B;QAAne,OYpjBO,W;  
;MZojBP,W;K;IAEJ,mC;MAOWB,kBAApB,qBAAC,SAAd,C;MAA4B,4B;MAA5B,OAA4C,oBY7jBrC,WZ6jBqC  
,C;K;IAEHd,iC;MAOWB,kBAApB,qBAAC,SAAd,C;MAA4B,+B;MAA5B,OYtkBO,W;K;0FZykBX,2B;MAKI,sB  
AAI,IAAK,MAAT,EAAGB,IAAK,OAArB,C;K;4FAGJ,yB;MAAA,gD;MAAA,mC;QAKI,kBAAO,KAAP,C;O;KA  
LJ,C;4FAQA,yB;MAAA,gD;MAAA,mC;QAKI,kBAAO,KAAP,C;O;KALJ,C;4FAQA,yB;MAAA,gD;MAAA,mC;

QAKI,kBAAO,KAAP,C;O;KALJ,C;4FAQA,0B;MAKI,yBAAO,GAAP,C;K;IAGJ,kC;MAOWB,kBAAf,aAAL,SA  
AK,C;MAAsCL,6B;MArCA,OAA+C,oBYxnBxC,WZwnBwC,C;K;IAEnD,mC;MAQwB,kBAAf,aAAL,SAAK,C;M  
AqCK,YAAL,gBAAK,O;MArCV,OAAgD,oBYloBzC,WZkoByC,C;K;IAEpD,mC;MAQwB,kBAAf,aAAL,SAAK,  
C;MAoCK,YAAL,gBAAK,O;MApCV,OAAgD,oBY5oBzC,WZ4oByC,C;K;IAEpD,mC;MAQwB,kBAAf,aAAL,S  
AAK,C;MAMCK,YAAL,gBAAK,O;MANCV,OAAgD,oBYtpBzC,WZspByC,C;K;4FAEpD,0B;MAMI,uBAAO,G  
AAP,C;K;8FAGJ,yB;MAAA,sD;MAAA,kC;QAMc,UAAV,SAAK,KAAK,EAAU,IAAV,C;O;KANd,C;8FASA,yB  
;MAAA,sD;MAAA,kC;QAMc,UAAV,SAAK,KAAK,EAAU,IAAV,C;O;KANd,C;8FASA,yB;MAAA,sD;MAAA,  
kC;QAMc,UAAV,SAAK,KAAK,EAAU,IAAV,C;O;KANd,C;IAUA,wC;MACsD,QAAM,cAAN,C;aACID,C;UAD  
kD,OAC7C,U;aACL,C;UAFkD,gB;;UAAA,OAG1C,S;;K;oF2KxwBZ,yB;MAAA,8D;MAAA,8B;MAAA,qC;QAU  
iC,Q;QAA7B,OAA2D,CAA9B,sEAA8B,eAAO,OAAP,C;O;KAV/D,C;wFAYA,yB;MAAA,8D;MAAA,8B;MAAA  
,sC;QASiC,Q;QAA7B,OAA2D,CAA9B,sEAA8B,oBAAU,QAAP,C;O;KAT/D,C;wFAWA,yB;MAAA,8D;MAAA,  
8B;MAAA,sC;QASiC,Q;QAA7B,OAA2D,CAA9B,sEAA8B,oBAAU,QAAP,C;O;KAT/D,C;4FAWA,8B;MAKI,S  
AAK,WAAL,OAAL,C;K;4FAGT,yB;MAAA,gD;MAAA,sC;QAKS,OAAL,SAAK,EAAO,QAAP,C;O;KALT,C;4F  
AQA,yB;MAAA,gD;MAAA,sC;QAKS,OAAL,SAAK,EAAO,QAAP,C;O;KALT,C;4FAQA,yB;MAAA,gD;MAA  
A,sC;QAKS,OAAL,SAAK,EAAO,QAAP,C;O;KALT,C;8FAQA,8B;MAKI,SAAK,cAAO,OAAP,C;K;8FAGT,yB;  
MAAA,sD;MAAA,sC;QAKS,UAAL,SAAK,EAAU,QAAP,C;O;KALT,C;8FAQA,yB;MAAA,sD;MAAA,sC;QAK  
S,UAAL,SAAK,EAAU,QAAP,C;O;KALT,C;8FAQA,yB;MAAA,sD;MAAA,sC;QAKS,UAAL,SAAK,EAAU,QA  
AV,C;O;KALT,C;IAQA,qC;MAIU,IAIe,I;MAHjB,kBADE,QACF,c;QAAiB,OAAO,yBAAO,QAAP,C;;QAEpB,a  
AAsB,K;QACT,0B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,oBAAL,IAAJ,CAAJ,C;YAAe,SAAS,I;;QAC5B,  
OAAO,M;;K;IAKnB,uC;MAKiB,Q;MADb,aAAsB,K;MACT,0B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAI,o  
BAAL,IAAJ,CAAJ,C;UAAe,SAAS,I;;MAE5B,OAAO,M;K;IAGX,uC;MAIL,OAAO,yBAAgB,OAAT,QAAS,CAA  
hB,C;K;IAGX,0C;MAIW,iBAAmB,gCAAT,QAAS,EAAgC,SAAhC,C;MAIHG,Q;MAKH7B,OAIH2D,CAA9B,sE  
AA8B,oBAAU,UAAV,C;K;IAqH/D,0C;MAIL,UAAmB,8BAAT,QAAS,C;MACnB,O5K4EwD,C4K5EjD,G5K4Ek  
D,U4K5EID,IAAoB,4BAAU,GAAV,C;K;IAG/B,0C;MAIL,OjL4oPO,EiL5oPA,QjLokPA,YAAQ,CAwER,CiL5oP  
A,IAAyB,4BAAmB,8BAAT,QAAS,CAAnB,C;K;IAGpC,0C;MAIW,iBAAmB,gCAAT,QAAS,EAAgC,SAAhC,C;  
MA7HG,Q;MA6H7B,OA7H2D,CAA9B,sEAA8B,oBAAU,UAAV,C;K;IAGl/D,0C;MAIL,IjL8nPO,EiL9nPH,QjLsj  
PG,YAAQ,CAwER,CiL9nPP,C;QACI,OAAO,4BAAmB,8BAAT,QAAS,CAAnB,C;;QAEp,OAAO,wB;K;IAGf,0C  
;MAIL,UAAmB,8BAAT,QAAS,C;MACnB,I5K4CwD,C4K5CpD,G5K4CqD,U4K5CzD,C;QACI,OAAO,4BAAU,  
GAAV,C;;QAEp,OAAO,wB;K;IAGf,kC;MACI,a5KqCwD,CAAC,mB;M4KpCzD,iB;MACA,OAAO,M;K;IAIX,2  
C;MAKkF,gCAAc,SAAd,EAAyB,IAAzB,C;K;IAEIF,2C;MAKkF,gCAAc,SAAd,EAAyB,KAAzB,C;K;IAEIF,sE;  
MACI,iBAaA,KAAb,C;M/JIjgB,kB+JmJX,oB;MACD,OAAO,qBAAP,C;QACI,IAAI,UAAU,kBAAV,6BAAJ,C;U  
ACI,oB;UACA,WAAS,I;;MAGrB,OAAO,Q;K;oFAIX,4B;MAM6D,kCAAS,KAAT,C;K;IAE7D,gC;MAKiD,IAAI,  
mBAAJ,C;QAAe,MAAM,2BAAUb,gBAAvB,C;;QAArB,OAAMe,2BAAS,CAAT,C;K;IAEpH,sC;MAKwD,OAAL  
,mBAAJ,GAAe,IAAf,GAAyB,2BAAS,CAAT,C;K;IAEjF,+B;MAKgD,IAAI,mBAAJ,C;QAAe,MAAM,2BAAUb,g  
BAAvB,C;;QAArB,OAAMe,2BAAS,2BAAT,C;K;IAEnH,qC;MAKuD,OAAL,mBAAJ,GAAe,IAAf,GAAyB,2BA  
AS,2BAAT,C;K;IAEhF,2C;MAK8E,kCAAc,SAAd,EAAyB,IAAzB,C;K;IAE9E,2C;MAK8E,kCAAc,SAAd,EAAy  
B,KAAzB,C;K;IAE9E,wE;MAEgB,UAGS,MAHT,EAcY,MAZ,EAc6B,M;MAfzC,IAAI,uCAAJ,C;QACI,OAAO  
C,cAA5B,sEAA4B,EAAc,SAAd,EAAyB,uBAAzB,C;MAExC,iBAAsB,C;MACD,oC;MAArB,qBAAkB,CAALB,m  
C;QACI,cAAc,sBAAK,SAAL,C;QACd,IAAI,UAAU,OAAP,MAAsB,uBAAIB,C;UACI,Q;QAEJ,IAAI,eAAc,SA  
AIB,C;UACI,sBAAK,UAAL,EAAmB,OAAnB,C;QAEJ,+B;;MAEJ,IAAI,aAAa,cAAjB,C;QACwB,oC;QAAiB,mB  
;QAArC,oE;UACI,2BAAS,WAAT,C;QAEJ,OAAO,I;;QAEp,OAAO,K;;K;ICjSf,wB;K;kCAEI,Y;MAA4B,sB;K;;I  
AMhC,wB;K;kCAEI,Y;MAA4B,mC;K;;IAMhC,yB;K;mCAEI,Y;MAA4B,uB;K;;IAMhC,uB;K;iCAEI,Y;MAA4B,  
qB;K;;IAMhC,wB;K;kCAEI,Y;MAA4B,sB;K;;IAMhC,yB;K;mCAEI,Y;MAA4B,uB;K;;IAMhC,0B;K;oCAEI,Y;M  
AA4B,wB;K;;IAMhC,2B;K;qCAEI,Y;MAA4B,yB;K;;ICzDc,wC;MAAkC,uB;MAAjC,0B;K;4FACpB,Y;MAAQ,O  
AAA,eAAS,K;K;iDACxC,iB;MAAkC,mCAAS,0BAAoB,KAApB,CAAT,C;K;;IAGT,gC;MAAyC,8B;MAAxC,0B  
;K;oFACH,Y;MAAQ,OAAA,eAAS,K;K;yCACxC,iB;MAAkC,mCAAS,0BAAoB,KAApB,CAAT,C;K;mCAEIC,Y  
;MAAuB,eAAS,Q;K;8CACHC,iB;MAAuC,OAAA,eAAS,kBAAS,0BAAoB,KAApB,CAAT,C;K;yCAEHd,0B;MA  
A8C,OAAA,eAAS,aAAI,0BAAoB,KAApB,CAAJ,EAAgC,OAAC,C;K;yCACvD,0B;MACI,eAAS,aAAI,2BAAq

B,KAARb,CAAJ,EAAiC,OAAjC,C;K;;IAIjB,+C;MACoB,Q;MAAA,kC;MAAhB,IAAa,CAAT,0BAAJ,C;QAAA,OAA2B,8BAAy,KAAZ,I;;QAAuB,MAAM,8BAA0B,mBAAgB,KAAhB,2BAA0C,gBAAG,2BAAH,CAA1C,OAA1B,C;K;IAE5D,gD;MACoB,Q;MAAA,qB;MAAhB,IAAa,CAAT,0BAAJ,C;QAAA,OAAsB,iBAAO,KAAP,I;;QAAkB,MAAM,8BAA0B,oBAAiB,KAAjB,2BAA2C,gBAAG,cAAH,CAA3C,OAA1B,C;K;IAGID,+B;MAK+C,gCAAqB,SAARb,C;K;IAE/C,iC;MAM6D,wBAAa,SAAb,C;K;;;IrkpC7D,oD;MAQuF,wC;K;IARvF,8CASI,Y;MAAuC,8B;K;IAT3C,gF;IsKa8G,wC;MAAA,mB;QAAE,kBAAS,aAAT,C;O;K;IAVhH,yB;MAUqG,oCAAS,sBAAT,C;K;IAErG,2B;MASI,eAAe,6B;MACf,oBAA0B,+BAAN,KAAM,EAAwC,QAAxC,EAA+D,QAA/D,C;MAC1B,OAAO,Q;K;IAc+B,yB;K;+CAoBtC,kC;MAOI,IAAI,uCAA0B,QAAS,UAAvC,C;QAAkD,M;MACID,OAAO,sBAAS,QAAS,WAA1B,e;K;+CAGX,kC;MAQqD,6BAAS,QAAS,WAA1B,e;K;;;IAyzD,mC;MAA2C,wB;MACvC,eAAoB,C;MACpB,mBAA4B,I;MAC5B,sBAAyC,I;MACzC,gBAAoC,I;K;gDAEpC,Y;MACI,OAAO,IAAP,C;QACI,QAAm,YAAN,C;eACI,C;YAAA,K;eACA,C;YACI,IAAI,kCAAE,UAAAnB,C;cACI,eAAQ,C;cACR,OAAO,I;;cAEP,sBA Ae,I;;YALvB,K;eAOA,C;YAAc,OAAO,K;eACrB,C;eAAA,C;YAAgC,OAAO,I;;YAC/B,MAAM,yB;;QAGlB,eAAQ,C;QACR,WAAW,4B;QACX,gBAAW,I;QACX,I5HpFR,oBDgDQ,W6HoCY,kb7HpCZ,CChDR,C;;K;6C4HwFA,Y;MACU,IASe,I;MATrB,QAAM,YAAN,C;aACI,C;aAAA,C;UAAsC,OAAO,qB;aAC7C,C;UACI,eAAQ,C;UACR,OAAO,kCAAE,O;aAE1B,C;UACI,eAAQ,C;UACR,aACa,mF;UACb,mBAAy,I;UACZ,OAAO,M;;UAEH,MAAM,yB;;K;uDAlTb,Y;MACI,IAAI,CAAC,cAAL,C;QAAGB,MAAM,6B;;QAA8B,OAAO,W;K;2DAG/D,Y;MAA4C,QAAM,YAAN,C;aACxC,C;UADwC,OAC1B,6B;aACd,C;UAFwC,OAExB,6BAAsB,sBAAtB,C;;UAFwB,OA GhC,6BAAsB,uCAAOc,YAA1D,C;;K;IAOqC,4E;MAAA,oB;QACzC,wCAAW,C;QAAX,OACA,yB;O;K;oDALR,+B;MACI,mBAAy,K;MACZ,eAAQ,C;MACR,OAA6C,0CAAtC,c;K;IAUsC,+E;MAAA,oB;QACzC,wCAAW,C;QAAX,OACA,yB;O;K;yDANR,kC;MACI,IAAI,CAAC,QAAS,UAAAd,C;QAAYB,M;MACzB,sBAAe,Q;MACf,eAAQ,C;MACR,OAA6C,6CAAtC,c;K;2DAMX,kB;M7HNO,Q;MADP,e6HSI,M7HTJ,C;MACO,Q6HQH,M7HRG,+D;M6HSH,eAAQ,C;K;kGAIR,Y;MAAQ,0C;K;;ItK/KhB,oD;MAQuF,wC;K;IARvF,8CASI,Y;MAAuC,8B;K;IAT3C,gF;sFAAA,yB;MAAA,kC;MAAA,0C;MAAA,kD;QAQuF,wC;O;MARvF,4CASI,Y;QAAuC,8B;O;MAT3C,8E;MAAA,2B;QAQuF,2C;O;KARvF,C;IAiBgE,+C;MAAA,mB;QAAE,sB;O;K;IALIE,kC;MAKuD,OAAkB,2CAAT,+BAAS,E;K;IAEzE,8B;MAK6D,OAAI,QdksPtD,YAAQ,CclsP0C,GAAwB,eAAxB,GAAsD,WAAT,QAAS,C;K;IAEnH,yB;MAG8C,kC;K;IAE9C,yB;MAAA,6B;K;uCACI,Y;MAA6C,kC;K;2CAC7C,a;MAA4B,kC;K;2CAC5B,a;MAA4B,kC;K;;IAHhC,qC;MAAA,oC;QAAA,mB;;MAAA,6B;K;oFAMA,yB;MAAA,2D;MAAA,4B;QAM4D,uCAAQ,e;O;KANpE,C;IAGB4F,mH;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,wC;MAAA,wD;MAAA,kC;K;;;kDAAA,Y;;;cACxF,eAAe,uBAAa,W;cAC5B,IAAI,QAAS,UAAb,C;gBACI,gB;gCAAA,sCAAS,QAAT,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBAEA,gB;gCAAA,sCAAS,iCAAT,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;;cAJJ,W;;cAAA,W;;;K;IADwF,gE;MAAA,yD;uBAAA,uG;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAP5F,4C;MAOmF,gBAAS,uCAAT,C;K;IAGBb,4B;MAAE,OAAA,EAAG,W;K;IAP3E,8B;MAO8D,4BAAQ,cAAR,C;K;IAUQ,8B;MAAE,OAAA,EAAG,W;K;IAR3E,8B;MAQ8D,4BAAQ,gBAAR,C;K;IAM1B,8B;MAAE,S;K;IAJtC,wC;MAEgB,Q;MADZ,IAAI,8CAAJ,C;QACI,OAA4C,CAApC,2EAAoC,kBAAQ,QAAR,C;;MAEHd,OAAO,uBAAmB,SAA nB,EAAyB,gBAAzB,EAAiC,QAAjC,C;K;IAGX,4B;MAYiB,Q;MAFb,YAAy.gB;MACZ,YAAy.gB;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,KAAM,WAAI,IAAK,MAAT,C;QACN,KAAM,WAAI,IAAK,OAAT,C;;MAEV,OAAO,UAAS,KAAT,C;K;IAGX,+B;MAQqD,6BAAS,4BAAT,C;K;IAW0B,+G;MAAA,wC;MAAA,6B;MAAAA,yB;MAAA,0C;MAAA,4C;MAAA,0B;MAAA,kC;K;;;mDAAA,Y;;;kCAC9D,0C;cACb,gB;;;cAAA,IAAO,iBT2FkD,US3FzD,C;gBAAA,gB;;;cACI,QAAQ,yBAAO,iBAAQ,iBAAO,KAAf,C;cACf,WAAkB,WAAP,iBAAO,C;cACIB,YAAgB,IAAI,iBAAO,KAAf,GAAqB,iBAAO,aAAI,CAAJ,EAAO,IAAP,CAA5B,GAA8C,I;cAC1D,gB;8BAAA,iCAAM,KAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cAJJ,gB;;;cAMJ,W;;;K;IAR+E,4D;MAAA,yD;uBAAA,mG;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAT/E,uC;MASmE,gBAAy,kCAAZ,C;K;IAkBhC,0D;MAE/B,wB;QAAA,WAAgC,I;MADhC,0B;MACA,0B;MACA,4B;K;IAGuC,0E;MAAA,oD;MACnC,gBAAe,iCAAS,W;MACxB,iBAAqB,E;MACrB,gBAAmB,I;K;oEAEnB,Y;MACI,OAAO,aAAS,UAAhB,C;QACI,WAAW,aAAS,O;QACpB,IAAI,wCAAU,IAAV,MAAmB,sCAAvB,C;UACI,gBAAW,I;UACX,iBAAy,C;UACZ,M;;MAGR,iBAAy,C;K;8DAGhB,Y;MASW,Q;MARP,IAAI,mBAAa,EAAjB,C;QACI,iB;MACJ,IAAI,mBAAa,CAAjB,C;QACI,MAAM,6B;MACV,aAAa,a;MACb,gBAAW,I;MACX,iBAAy,E;MAEZ,OAAO,yE;K;IEAGX,Y;MACI,IAAI,mBAAa,EAAjB,C;QACI,iB;MACJ,OAAO,mBAAa,C;K;;2CAhC5B,Y;MAAuC,yD;K;;IA2C3C,qD;MAAY,0B;MAAM

C,gC;K;IACJ,gF;MAAA,0D;MACnC,gBA Ae,oCAAS,W;K;IEACxB,Y;MACI,OAAO,6CAAY,aAAS,OAArB,C;K ;oEAGX,Y;MACI,OAAO,aAAS,U;K;;8CAPxB,Y;MAAuC,4D;K;qDAWvC,oB;MACI,OAAO,uBAA4B,eAA5B,E AAsC,kBAAtC,EAAMd,QAAnD,C;K;;IAUf,4D;MAAY,0B;MAAmC,gC;K;IACJ,8F;MAAA,wE;MACnC,gBA Ae ,2CAAS,W;MACxB,aAAY,C;K;wEACZ,Y;MAC0C,Q;MAAtC,OAAO,oDAAY,oBAAmB,iBAAnB,EAAMb,yBA AnB,QA AZ,EAAYC,aAAS,OAAID,C;K;2EAGX,Y;MACI,OAAO,aAAS,U;K;;qDARxB,Y;MAAuC,mE;K;;IAkB3 C,oC;MAAY,0B;K;IAC6C,wE;MACjD,gBA Ae,gCAAS,W;MACxB,aAAY,C;K;6DACZ,Y;MAC2C,Q;MAAvC,O AAO,iBAAa,oBAAmB,iBAAnB,EAAMb,yBAAnB,QAAb,EAA0C,aAAS,OAAAnD,C;K;gEAGX,Y;MACI,OAAO, aAAS,U;K;;0CARxB,Y;MAAqD,wD;K;;IAmBzD,0D;MACI,4B;MACA,4B;MACA,4B;K;IAEuC,sE;MAAA,gD; MACnC,iBAAgB,gCAAU,W;MAC1B,iBAAgB,gCAAU,W;K;4DAC1B,Y;MACI,OAAO,sCAAU,cAAU,OAApB, EAA4B,cAAU,OAAAtC,C;K;+DAGX,Y;MACI,OAAO,cAAU,UAAV,IAAuB,cAAU,U;K;;yCARhD,Y;MAAuC,uD ;K;;IAc3C,6D;MACI,0B;MACA,gC;MACA,0B;K;IAEuC,4E;MAAA,sD;MACnC,gBA Ae,kCAAS,W;MACxB,oB AAiC,I;K;+DAEjC,Y;MACI,IAAI,CAAC,2BAAL,C;QACI,MAAM,6B;MACV,OAAO,gCA Ae,O;K;kEAG1B,Y; MACI,OAAO,2B;K;+EAGX,Y;MACQ,Q;MAAJ,IAAI,iEAA2B,KAA/B,C;QACI,oBA Ae,I;MAEnB,OAAO,yBAA P,C;QACI,IAAI,CAAC,aAAS,UAAAd,C;UACI,OAAO,K;;UAEP,cAAc,aAAS,O;UACvB,uBAAuB,wCAAS,2CAA Y,OAAZ,CAAT,C;UACvB,IAAI,gBAAiB,UAArB,C;YACI,oBA Ae,gB;YACf,OAAO,I;;;MAInB,OAAO,I;K;;4C A9Bf,Y;MAAuC,0D;K;;IAoC9B,6I;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,4C;MAAA,kD;MAAA,gD;MAAA ,wB;MAAA,yB;MAAA,kC;K;;;yDAAA,Y;;;kBAGyC,I;ICAFIC,C;cACI,sD;cAAhB,gB;;;cAAA,KAAgB,yBAA hB,C;gBAAA,gB;;;cAAGB,oC;cACZ,aAAa,6BAAU,oBAAmB,uBAAnB,EAAMb,+BAAnB,QA AV,EAAuC,OAA vC,C;cACb,gB;8BAAA,sCAAS,4BAAS,MAAT,CAAT,O;kBAAA,2C;uBAAA,yB;cAAA,Q;cAFJ,gB;;;cAIJ,W;;; ;K;IANS,0F;MAAA,yD;uBAAA,iI;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IADb,wD;MACI,gBAAS,kDAAT, C;K;;;IAoByB,qD;MACzB,0B;MACA,8B;MACA,0B;MC3TA,IAAI,ED+TQ,qBAAc,CC/TtB,CAAJ,C;QACI,cD8 T2B,+CAA4C,iB;QC7TvE,MAAM,gCAAYB,OAAQ,WAAjC,C;;MAFV,IAAI,EDgUQ,mBAAy,CChUpB,CAAJ, C;QACI,gBD+TyB,6CAA0C,e;QC9TnE,MAAM,gCAAYB,SAAQ,WAAjC,C;;MAFV,IAAI,EDiUQ,mBAAy,iBcj UpB,CAAJ,C;QACI,gBDgUkC,0DAAuD,eAAvD,WAAmE,iB;QC/TrG,MAAM,gCAAYB,SAAQ,WAAjC,C;;K;sF DkUa,Y;MAAQ,yBAAW,iBAAX,I;K;yCAE/B,a;MAAYC,OAAI,KAAK,YAAT,GAAgB,eAAhB,GAAqC,gBAAy ,eAAZ,EAAsB,oBAAa,CAAb,IAAtB,EAAsC,eAAtC,C;K;yCAC9E,a;MAAYC,OAAI,KAAK,YAAT,GAAgB,IAA hB,GAA0B,gBAAy,eAAZ,EAAsB,iBAAtB,EAAC,oBAAa,CAAb,IAAI,C;K;IAEzC,8D;MAAA,wC;MAEtB,g BA Ae,2BAAS,W;MACxB,gBA Ae,C;K;0DAEf,Y;MAEI,OAAO,gBAAW,kCAAX,IAAYB,aAAS,UAAzC,C;QACI ,aAAS,O;QACT,qC;;K;2DAIR,Y;MACI,a;MACA,OAAQ,gBAAW,gCAAZ,IAAYB,aAAS,U;K;wDAG7C,Y;MAC I,a;MACA,IAAI,iBAAy,gCAAhB,C;QACI,MAAM,6B;MACV,qC;MACA,OAAO,aAAS,O;K;qCAvBxB,Y;MAA 0B,mD;K;;IAgCA,uC;MAC1B,0B;MACA,oB;MC3WA,IAAI,ED+WQ,gBAAS,CC/WjB,CAAJ,C;QACI,cD8WsB, yCAAsC,YAAAtC,M;QC7WtB,MAAM,gCAAYB,OAAQ,WAAjC,C;;K;0CDgXV,a;MAAYC,OAAI,KAAK,YAAT, GAAgB,eAAhB,GAAqC,gBAAy,eAAZ,EAAsB,CAAtB,EAAYB,YAAzB,C;K;0CAC9E,a;MAAYC,OAAI,KAAK, YAAT,GAAgB,IAAhB,GAA0B,iBAAa,eAAb,EAAuB,CAAvB,C;K;IAE5B,gE;MACnC,YAAW,yB;MACX,gBA Ae,4BAAS,W;K;yDAExB,Y;MACI,IAAI,cAAQ,CAAZ,C;QACI,MAAM,6B;MACV,6B;MACA,OAAO,aAAS,O; K;4DAGpB,Y;MACI,OAAO,YAAO,CAAP,IAAY,aAAS,U;K;;sCAZpC,Y;MAAuC,oD;K;;IASB3C,gD;MACI,0B; MACA,4B;K;IAEuC,0E;MAAA,oD;MACnC,gBA Ae,iCAAS,W;MACxB,iBAAqB,E;MACrB,gBAAMb,I;K;oEAE nB,Y;MACI,IAAI,aAAS,UAAb,C;QACI,WAAW,aAAS,O;QACpB,IAAI,wCAAU,IAAV,CAAJ,C;UACI,iBAAy, C;UACZ,gBAAW,I;UACX,M;;;MAGR,iBAAy,C;K;8DAGhB,Y;MAMiB,Q;MALb,IAAI,mBAAa,EAAjB,C;QAC I,iB;MACJ,IAAI,mBAAa,CAAjB,C;QACI,MAAM,6B;MACV,aACa,gF;MAGb,gBAAW,I;MACX,iBAAy,E;MA CZ,OAAO,M;K;IEAGX,Y;MACI,IAAI,mBAAa,EAAjB,C;QACI,iB;MACJ,OAAO,mBAAa,C;K;;2CAIC5B,Y;MA AuC,yD;K;;IA2Cb,uC;MAC1B,0B;MACA,oB;MC5bA,IAAI,ED+bQ,gBAAS,CC/bjB,CAAJ,C;QACI,cD8bsB,yC AAsC,YAAAtC,M;QC7btB,MAAM,gCAAYB,OAAQ,WAAjC,C;;K;0CDgcV,a;MItXO,SJsXmC,eAAQ,CAAR,I;M AAD,OAA4B,KAAK,CAAT,GAAY,yBAAZ,GAAuC,iBAAa,eAAb,EAAuB,EAAvB,C;K;0CACxG,a;MIvXO,SJu XmC,eAAQ,CAAR,I;MAAD,OAA4B,KAAK,CAAT,GAAY,yBAAZ,GAAuC,gBAAy,eAAZ,EAAsB,YAAAtB,EA A6B,EAA7B,C;K;IAEjE,gE;MACnC,gBA Ae,4BAAS,W;MACxB,YAAW,yB;K;2DAEX,Y;MAEI,OAAO,YAAO, CAAP,IAAY,aAAS,UAA5B,C;QACI,aAAS,O;QACT,6B;;K;yDAIR,Y;MACI,a;MACA,OAAO,aAAS,O;K;4DAG pB,Y;MACI,a;MACA,OAAO,aAAS,U;K;;sCAnBxB,Y;MAAuC,oD;K;;IA6B3C,gD;MACI,0B;MACA,4B;K;IAGu

C,0E;MAAA,oD;MACnC,gBA Ae,iCAAS,W;MACxB,iBAAqB,E;MACrB,gBAAmB,I;K;gEAEnB,Y;MACI,OAA  
O,aAAS,UAAhB,C;QACI,WAAW,aAAS,O;QACpB,IAAI,CAAC,wCAAU,IAAV,CAAL,C;UACI,gBAAW,I;UAC  
X,iBAAY,C;UACZ,M;;;MAGR,iBAAY,C;K;8DAGhB,Y;MAMqB,Q;MALjB,IAAI,mBAAa,EAAjB,C;QACI,a;M  
AEJ,IAAI,mBAAa,CAAjB,C;QACI,aACa,gF;QACb,gBAAW,I;QACX,iBAAY,C;QACZ,OAAO,M;;;MAEX,OAA  
O,aAAS,O;K;iEAGpB,Y;MACI,IAAI,mBAAa,EAAjB,C;QACI,a;MACJ,OAAO,mBAAa,CAAb,IAAkB,aAAS,U;  
K;;2CAIC1C,Y;MAAuC,yD;K;;IAuCN,+C;MAAC,sB;MAAiC,gC;K;0CACnE,Y;MAAuC,4BAAiB,aAAO,WAAx  
B,EAAoC,kBAApC,C;K;;IAGP,+C;MAAuE,2B;MAAtE,sB;MAAiC,gC;MACIE,kBAAuB,c;K;6CAEvB,Y;MACI,  
OAAO,aAAO,UAAAd,C;QACI,WAAW,aAAO,O;QACIB,UAAU,mBAAAY,IAAZ,C;QAEV,IAAI,eAAS,WAAI,GA  
AJ,CAAb,C;UACI,mBAAQ,IAAR,C;UACA,M;;;MAIR,W;K;;IAKgC,0D;MAAC,wC;MAAuC,kC;K;IACrC,0E;M  
AAA,oD;MACnC,gBAAmB,I;MACnB,iBAAqB,E;K;0EAERB,Y;MACI,gBA Ae,mBAAa,EAAjB,GAAqB,+CAAr  
B,GAA4C,2CAAA,4BAAb,C;MACvD,iBAAgB,qBAAJ,GAAsB,CAAtB,GAA6B,C;K;8DAG7C,Y;MAMiB,Q;MA  
Lb,IAAI,iBAAY,CAAhB,C;QACI,iB;MAEJ,IAAI,mBAAa,CAAjB,C;QACI,MAAM,6B;MACV,aAAa,8D;MAEb,i  
BAAY,E;MACZ,OAAO,M;K;iEAGX,Y;MACI,IAAI,iBAAY,CAAhB,C;QACI,iB;MACJ,OAAO,mBAAa,C;K;;2C  
Ax5B,Y;MAAuC,yD;K;;IA6B3C,kC;MAWI,OAAW,iDAAJ,GAAwC,SAAxC,GAakD,4BAAwB,SAAxB,C;K;I  
AelB,uD;MAAA,qB;QAAE,6B;O;K;IAX7C,wC;MAWI,OAA2D,cAApD,sBAakB,YAAIB,EAAgC,qCAAhC,CA  
AoD,C;K;IAqBrC,iD;MAAA,mB;QAAE,mB;O;K;IAIB5B,gD;MAeI,0AAI,YAAJ,GACI,2BADJ,GAGI,sBAakB,  
+BAAIB,EAA4B,YAA5B,C;K;IAER,wD;MAcI,6BAakB,YAAIB,EAAgC,YAAhC,C;K;IPxpBJ,oB;MAAA,wB;M  
ACI,8C;K;gCAEA,iB;MAA4C,oCAAmB,KAAM,U;K;kCACrE,Y;MAA+B,Q;K;kCAC/B,Y;MAAkC,W;K;gFAE  
X,Y;MAAQ,Q;K;iCAC/B,Y;MAAkC,W;K;wCACIC,mB;MAAmD,Y;K;6CACnD,oB;MAAmE,OAAA,QAAS,U;  
K;kCAE5E,Y;MAA6C,kC;K;uCAE7C,Y;MAAiC,6B;K;;IADrC,gC;MAAA,+B;QAAA,c;;MAAA,wB;K;IAkBA,o  
B;MAIoC,6B;K;IAEpC,2B;MAMmD,OAAI,QAAS,OAAT,GAAgB,CAApB,GAAgC,MAAT,QAAS,CAAhC,GA  
A6C,U;K;IFAeHg,yB;MAAA,mD;MAAA,mB;QAKwC,iB;O;KALxC,C;6FAOA,yB;MAAA,uE;MAAA,mB;QAQ  
sD,2B;O;KARtD,C;IAUA,kC;MAKiE,OAAS,aAAT,QAAS,EAAa,qBAAc,YAAY,QAAS,OAARB,CAAd,CAAb,C;  
K;uFAE1E,yB;MAAA,2D;MAAA,mB;QAGgD,qB;O;KAHhD,C;IAKA,+B;MAC2D,OAAS,aAAT,QAAS,EAAa,e  
AAQ,YAAY,QAAS,OAARB,CAAR,CAAb,C;K;2FAEpE,yB;MAAA,uE;MAAA,mB;QAMwD,2B;O;KANxD,C;I  
AQA,iC;MAKmE,OAAS,aAAT,QAAS,EAAa,qBAAc,YAAY,QAAS,OAARB,CAAd,CAAb,C;K;IAE5E,+B;MAM  
yD,OAAI,eAAJ,GAAqB,MAAM,OAAN,CAARB,GAAyC,U;K;IAEIG,kC;MAQI,OAAgB,gBAAT,QAAS,EAAgB,  
sBAAhB,C;K;sFAGpB,yB;MetBA,uE;MfsBA,gC;QelB8B,gBAAnB,oB;QfqCiB,aWhDxB,W;QXgDA,OW/CO,SI  
UwC,Q;O;KfkBnD,C;wFA2BA,yB;Me1CA,wE;Mf0CA,0C;QetCsC,gBAA3B,mBf6DiB,Qe7DjB,C;Qf6D2B,aW/E  
IC,W;QX+EA,OW9EO,SliBgD,Q;O;KfsC3D,C;sFAGCA,yB;MAAA,mD;MAAA,4B;QAEkD,uCAAQ,U;O;KAF1  
D,C;IAIA,wC;MAAgD,QAAM,cAAN,C;aAC5C,C;UAD4C,OACvC,U;aACL,C;UAF4C,0AEvC,MAAM,oBAAW  
,OAAjB,C;;UAFuC,OAGpC,S;;K;IOrkZ,oD;MAQuF,wC;K;IARvF,8CASI,Y;MAAuC,8B;K;IAT3C,gF;IuKLA,yC  
;MtK4BI,IAAI,EsK3BI,OAAO,CAAP,IAAY,OAAO,CtK2BvB,CAAJ,C;QACI,csK3BI,aAAJ,GACI,yEADJ,GAGI  
,8C;QtKyBJ,MAAM,gCAAYB,OAAQ,WAAjC,C;;K;IsKnBM,mI;MAAA,mB;QAAE,wBAAiB,gCAAjB,EAA6B,  
YAA7B,EAAmC,YAAAnC,EAAyC,sBAAzC,EAAyD,mBAAzD,C;O;K;IAFtB,gF;MACI,oBAAoB,IAApB,EAA0B,  
IAA1B,C;MACA,oCAAgB,6EAAhB,C;K;IAKyB,yL;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,wC;MAAA,wC;  
MAAA,gD;MAAA,sD;MAAA,4D;MAAA,wB;MAAA,0B;MAAA,uB;MAAA,0B;MAAA,wB;MAAA,qB;MAAA,  
4B;MAAA,kC;K;;;2DAAA,Y;;;cACrB,4BAAiC,eAAL,uBAAK,EAAa,IAAb,C;+BACvB,0BAAO,uBAAP,I;cAC  
V,IAAI,kBAAO,CAAX,C;oCACiB,iBAAa,qBAAb,C;kCACF,C;gBACD,6C;gBAAV,iB;;;sCAaa,gBAAc,qBAAd,  
C;gBACH,+C;gBAAV,gB;;;cAAA,KAAU,2BAAV,C;gBAAA,gB;;;cAAU,kC;cACN,mBAAO,WAAI,GAAJ,C;  
cACP,IAAI,mBAAO,SAAX,C;gBACI,IAAI,mBAAO,KAAP,GAAC,uBAAIB,C;kBAA0B,sBAAS,mBAAO,kBAA  
uB,uBAAvB,C;kBAA8B,gB;;;kBAAxE,gB;;;gBADJ,gB;;;cAGI,gB;8BAAA,iCAAU,8BAAJ,GAAiB,mBAAjB,G  
AA6B,iBAAU,mBAAV,CAAnC,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cACA,mBAAO,qBAAY,uBAAZ,C;cAJX,g  
B;;;cAFJ,gB;;;cASA,IAAI,iCAAJ,C;gBACI,gB;;;gBADJ,iB;;;cACI,IAAO,mBAAO,KAAd,IAAqB,uBAARb,C;gB  
AAA,gB;;;cACI,gB;8BAAA,iCAAU,8BAAJ,GAAiB,mBAAjB,GAA6B,iBAAU,mBAAV,CAAnC,O;kBAAA,2C;  
uBAAA,yB;cAAA,Q;;cACA,mBAAO,qBAAY,uBAAZ,C;cAFX,gB;;;cAIA,IhL8K4C,CgL9KxC,mBhL8KyC,Ug  
9K7C,C;gBAAyB,iB;gCAAA,iCAAM,mBAAN,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBAAzB,iB;;;cAjCR,W;;  
cA4BI,iB;;;cA1BJ,iB;;;cAGI,KAAU,yBAAV,C;gBAAA,iB;;;6BAAU,sB;cACN,IAAI,kBAAO,CAAX,C;gBAAgB

,oCAAQ,CAAR,I;gBAAW,iB;;;gBAA3B,iB;;;;cACA,iBAAO,WAAL,YAAJ,C;cACP,IAAI,iBAAO,KAAP,KAAe,  
uBAAnB,C;gBACI,iB;gCAAA,iCAAM,iBAAN,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBADJ,iB;;;;cAEI,IAAI,8  
BAAJ,C;gBAAiB,iBAAO,Q;;gBAAa,oBAAS,iBAAU,uBAAV,C;cAC9C,kBAAO,c;cAHX,iB;;;cAHJ,iB;;;cASA,Ih  
LiMgD,CgLjM5C,iBhLiM6C,UgLjMjD,C;gBACI,IAAI,qCAAKB,iBAAO,KAAP,KAAe,uBAArC,C;kBAA2C,iB;k  
CAAA,iCAAM,iBAAN,O;sBAAA,2C;2BAAA,yB;kBAAA,Q;;kBAA3C,iB;;;;gBADJ,iB;;;;cAdJ,W;;cAcI,iB;;;cA  
ZJ,iB;;;cAkCJ,W;;;;K;IArCyB,sI;MAAA,yD;uBAAA,6K;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAF7B,6E;  
MACI,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,2B;MACHC,OAAO,WAaKB,0EAAIB,C;K;IAwCwB,6B;MA  
A8B,uB;MAA7B,kB;MACHC,mBAA6B,C;MAC7B,eAAyB,C;K;2CAEzB,8B;MACI,+DAaKB,SAaIB,EAA6B,O  
AA7B,EAAc,WAAK,KAA3C,C;MACA,mBAAiB,S;MACjB,eAAa,UAAU,SAAV,I;K;0CAGjB,iB;MACI,+DAA  
kB,KAAIB,EAAyB,YAAzB,C;MAEA,OAAO,wBAAK,mBAAy,KAAZ,IAAL,C;K;qFAGY,Y;MAAQ,mB;K;;IAS  
R,wC;MAAqD,uB;MAApD,sB;MtKrDxB,IAAI,EsKuDQ,cAAc,CtKvDtB,CAAJ,C;QACI,csKsD2B,wE;QtKrD3B,  
MAAM,gCAAyB,OAAQ,WAAjC,C;;MAFV,IAAI,EsKwDQ,cAAc,aAAO,OtKxD7B,CAAJ,C;QACI,gBsKuDqC,  
wFAA+E,aAAO,O;QtKtD3H,MAAM,gCAAyB,SAAQ,WAAjC,C;;MsK2DV,kBAAuB,aAAO,O;MAC9B,oBAA8  
B,C;MAE9B,sBAAYB,U;K;kFAAZB,Y;MAAA,0B;K,OAAA,gB;MAAA,0B;K;uCAGA,iB;MAGW,Q;MAFP,+DA  
AkB,KAAIB,EAAyB,SAAZB,C;MAEA,OAAO,sBAAmGmC,CAnG5B,iBAmG6B,GAnGV,KAmGU,IAAD,IAAa,e  
AAb,IAnGnC,4D;K;kCAGX,Y;MAAe,qBAAQ,e;K;IAEgB,4D;MAAA,sC;MAAS,2B;MAC5C,eAAoB,oB;MACp  
B,eAAoB,4B;K;8DAEpB,Y;MAKGB,Q;MAJZ,IAAI,iBAAS,CAAb,C;QACI,W;;QAGA,mBAAQ,sCAAQ,YAAP,4  
DAAR,C;QACA,eAoFkC,CaPFIb,YAoF2B,GaPfb,CAoFa,IAAD,IAAa,+BAAb,I;QAnFIC,mC;;K;;oCAXZ,Y;M  
AAuC,kD;K;2CAgBvC,iB;MAGIE,UAQ1C,MAROC,EAe1C,MAf0C,EAqBtD,M;MatBP,aACQ,KAAM,OAAO,G  
AAa,IAAK,KAAIB,GAaK,UAAO,KAAM,EAAO,IAAK,KAAZ,CAAI,GAAYD,kD;MAE7D,WAAW,IAAK,K;  
MAEhB,WAAW,C;MACX,UAAU,iB;MAEV,OAAO,OAAO,IAAP,IAAe,MAAM,eAA5B,C;QACI,OAAO,IAAP,I  
AAe,wBAAO,GAAP,gE;QACf,mB;QACA,iB;;MAGJ,MAAM,C;MACN,OAAO,OAAO,IAAd,C;QACI,OAAO,IA  
AP,IAAe,wBAAO,GAAP,gE;QACf,mB;QACA,iB;;MAEJ,IAAI,MAAO,OAAP,GAAC,IAAK,KAAvB,C;QAA6B,  
OAAO,IAAK,KAAZ,IAAoB,I;MAEjD,OAAO,uD;K;mCAGX,Y;MACI,OAAO,qBAAQ,gBAAa,SAAb,OAAR,C;  
K;4CAGX,uB;MAKI,kBAAoD,eAAjC,mBAAy,mBAAa,CAAZB,IAA8B,CAA9B,IAAiC,EAAa,WAAb,C;MACp  
D,gBAAoB,sBAAC,CAaIB,GAA4B,UAAP,aAAO,EAAO,WAAP,CAA5B,GAAqD,qBAAQ,gBAAa,WAAb,OAA  
R,C;MACrE,OAAO,eAAW,SAAX,EAA5B,SAAtB,C;K;qCAGX,mB;MAII,IAAI,aAAJ,C;QACI,MAAM,6BAAsB,  
qBAAtB,C;;MAGV,cA6B0C,CA7BnC,iBA6BoC,GA7BjB,SA6BiB,IAAD,IAAa,eAAb,IA7B1C,IAAmC,O;MACn  
C,6B;K;+CAGJ,a;MtKhJA,IAAI,EsKoJQ,KAAK,CtKpJb,CAAJ,C;QACI,csKmJkB,wC;QtKIJIB,MAAM,gCAAyB  
,OAAQ,WAAjC,C;;MAFV,IAAI,EsKqJQ,KAAK,StKrJb,CAAJ,C;QACI,gBsKoJqB,wEAA8D,S;QtKnJnF,MAAM  
,gCAAyB,SAAQ,WAAjC,C;;MsKqJN,IAAI,IAAI,CAAR,C;QACI,YAAy,iB;QACZ,UAGBsC,CAhB5B,KAgB6B,  
GAhBf,CagBe,IAAD,IAAa,eAAb,I;QAdtC,IAAI,QAAQ,GAAZ,C;UACW,OAAP,aAAO,EAAK,IAAL,EAAW,K  
AAX,EAAKB,eAAIB,C;UACA,OAAP,aAAO,EAAK,IAAL,EAAW,CAAX,EAAc,GAAd,C;;UAEA,OAAP,aAAO,  
EAAK,IAAL,EAAW,KAAx,EAAKB,GAAIB,C;;QAGX,oBAAa,G;QACb,wBAAQ,CAAR,I;;K;qCAKR,wB;MAC  
8C,QAAC,YAAO,CAAP,IAAD,IAAa,eAAb,I;K;;IA9G9C,0C;MAAA,oD;MAA6B,uBAAK,gBAAmB,QAAnB,O  
AAL,EAAmC,CAAnC,C;MAA7B,Y;K;ICvFJ,0C;MAII,QAAQ,I;MACR,QAAQ,K;MACR,YAAy,kBAAM,CAA  
C,OAAO,KAAP,IAAD,IAAiB,CAAjB,IAAN,C;MACZ,OAAO,KAAK,CAAZ,C;QACI,OpL+B4E,0BoL/BrE,kBA  
AM,CAAN,CpL0Q2B,KAAI,GAAiB,GA308B,EoL/B1D,KpL0QgB,KAAI,GAAiB,GA308B,CoL/BrE,IAAP,C;  
UACI,a;;QACJ,OpL6B4E,0BoL7BrE,kBAAM,CAAN,CpLwQ2B,KAAI,GAAiB,GA308B,EoL7B1D,KpLwQgB,  
KAAI,GAAiB,GA308B,CoL7BrE,IAAP,C;UACI,a;;QACJ,IAAI,KAAK,CAAT,C;UACI,UAAU,kBAAM,CAAN,  
C;UACV,kBAAM,CAAN,EAAW,kBAAM,CAAN,CAAX,C;UACA,kBAAM,CAAN,EAAW,GAAX,C;UACA,a;U  
ACA,a;;MAGR,OAAO,C;K;IAGX,uC;MAGI,YAAy,aAAU,KAAV,EAAiB,IAAjB,EAAuB,KAAvB,C;MACZ,IA  
AI,QAAO,QAAQ,CAAR,IAAP,CAAJ,C;QACI,UAAU,KAAV,EAAiB,IAAjB,EAAuB,QAAQ,CAAR,IAAvB,C;M  
ACJ,IAAI,QAAQ,KAAZ,C;QACI,UAAU,KAAV,EAAiB,KAAjB,EAAwB,KAAxB,C;K;IAGR,0C;MAII,QAAQ,I;  
MACR,QAAQ,K;MACR,YAAy,kBAAM,CAAC,OAAO,KAAP,IAAD,IAAiB,CAAjB,IAAN,C;MACZ,OAAO,K  
AAK,CAAZ,C;QACI,OILM6E,0BkLNtE,kBAAM,CAAN,CIL002B,KAAI,GAAiB,KApO+B,EkLN3D,KIL00gB,  
KAAI,GAAiB,KApO+B,CkLNtE,IAAP,C;UACI,a;;QACJ,OIL6E,0BkLJtE,kBAAM,CAAN,CILw02B,KAAI,G  
AAiB,KApO+B,EkLJ3D,KILwOgB,KAAI,GAAiB,KApO+B,CkLJtE,IAAP,C;UACI,a;;QACJ,IAAI,KAAK,CAA



T,C;UACI,UAAU,kBAAM,CAAN,C;UACV,kBAAM,CAAN,EAAW,kBAAM,CAAN,CAAX,C;UACA,kBAAM,CAAN,EAAW,GAAX,C;UACA,a;UACA,a;;MAGR,OAAO,C;K;IAGX,yC;MAGI,YAAY,aAAU,KAAV,EAAiB,IAAjB,EAAuB,KAAvB,C;MACZ,IAAI,QAAO,QAAQ,CAAR,IAAP,CAAJ,C;QACI,YAAU,KAAV,EAAiB,IAAjB,EAAuB,QAAQ,CAAR,IAAvB,C;MACJ,IAAI,QAAQ,KAAZ,C;QACI,YAAU,KAAV,EAAiB,KAAjB,EAAwB,KAAxB,C;K;IAGR,0C;MAII,QAAQ,I;MACR,QAAQ,K;MACR,YAAY,kBAAM,CAAC,OAAO,KAAP,IAAD,IAAiB,CAAjB,IAAN,C;MACZ,OAAO,KAAK,CAAZ,C;QACI,OnLnB8D,YmLmBvD,kBAAM,CAAN,CnLnBwE,KA AjB,EmLmB5C,KnLnByE,KAA7B,CmLmBvD,IAAP,C;UACI,a;;QACJ,OnLrB8D,YmLqBvD,kBAAM,CAAN,Cn LrBwE,KAAjB,EmLqB5C,KnLrByE,KAA7B,CmLqBvD,IAAP,C;UACI,a;;QACJ,IAAI,KAAK,CAAT,C;UACI,U AAU,kBAAM,CAAN,C;UACV,kBAAM,CAAN,EAAW,kBAAM,CAAN,CAAX,C;UACA,kBAAM,CAAN,EAA W,GAAX,C;UACA,a;UACA,a;;MAGR,OAAO,C;K;IAGX,yC;MAGI,YAAY,aAAU,KAAV,EAAiB,IAAjB,EAAu B,KAAvB,C;MACZ,IAAI,QAAO,QAAQ,CAAR,IAAP,CAAJ,C;QACI,YAAU,KAAV,EAAiB,IAAjB,EAAuB,QA AQ,CAAR,IAAvB,C;MACJ,IAAI,QAAQ,KAAZ,C;QACI,YAAU,KAAV,EAAiB,KAAjB,EAAwB,KAAxB,C;K;I AGR,0C;MAII,QAAQ,I;MACR,QAAQ,K;MACR,YAAY,kBAAM,CAAC,OAAO,KAAP,IAAD,IAAiB,CAAjB,IA AN,C;MACZ,OAAO,KAAK,CAAZ,C;QACI,OIK5C+D,akK4CxD,kBAAM,CAAN,CIK5C0E,KAAiB,EkK4C7C, KIK5C2E,KAA9B,CkK4CxD,IAAP,C;UACI,a;;QACJ,OIK9C+D,akK8CxD,kBAAM,CAAN,CIK9C0E,KAAiB,Ek K8C7C,KIK9C2E,KAA9B,CkK8CxD,IAAP,C;UACI,a;;QACJ,IAAI,KAAK,CAAT,C;UACI,UAAU,kBAAM,CAA N,C;UACV,kBAAM,CAAN,EAAW,kBAAM,CAAN,CAAX,C;UACA,kBAAM,CAAN,EAAW,GAAX,C;UACA,a ;UACA,a;;MAGR,OAAO,C;K;IAGX,yC;MAGI,YAAY,aAAU,KAAV,EAAiB,IAAjB,EAAuB,KAAvB,C;MACZ, IAAI,QAAO,QAAQ,CAAR,IAAP,CAAJ,C;QACI,YAAU,KAAV,EAAiB,IAAjB,EAAuB,QAAQ,CAAR,IAAvB,C ;MACJ,IAAI,QAAQ,KAAZ,C;QACI,YAAU,KAAV,EAAiB,KAAjB,EAAwB,KAAxB,C;K;IAKR,gD;MAI6E,UA AU,KAAV,EAAiB,SAAjB,EAA4B,UAAU,CAAV,IAA5B,C;K;IAC7E,gD;MAC6E,YAAU,KAAV,EAAiB,SAAj B,EAA4B,UAAU,CAAV,IAA5B,C;K;IAC7E,gD;MAC6E,YAAU,KAAV,EAAiB,SAAjB,EAA4B,UAAU,CAAV,I AA5B,C;K;IAC7E,gD;MAC6E,YAAU,KAAV,EAAiB,SAAjB,EAA4B,UAAU,CAAV,IAA5B,C;K;IrK9I7E,0C;M F0BI,IAAI,EEjBI,SAAU,OAAV,GAaiB,CfiBrB,CAAJ,C;QACI,cAda,qB;QAeb,MAAM,gCAAYB,OAAQ,WAAj C,C;;MEIBV,OAAO,oBAaOb,CAApB,EAAuB,CAAvB,EAA0B,SAA1B,C;K;IAGX,8C;MACe,Q;MAAX,wBAA W,SAAx,gB;QAAW,SAAA,SAAx,M;QACI,SAAS,GAAG,CAAH,C;QACT,SAAS,GAAG,CAAH,C;QACT,WA AW,cAAc,EAAAd,EAAkB,EAAiB,C;QACX,IAAI,SAAQ,CAAZ,C;UAAe,OAAO,I;;MAE1B,OAAO,C;K;sGAGX, yB;MAAA,8D;MAAA,iC;QASI,OAAO,cAAc,SAAS,CAAT,CAAd,EAA2B,SAAS,CAAT,CAA3B,C;O;KATX,C; sGAYA,sC;MASI,OAAO,UAAW,SAAQ,SAAS,CAAT,CAAR,EAAqB,SAAS,CAAT,CAArB,C;K;IAatB,6B;MA WY,Q;MALR,IAAI,MAAM,CAAV,C;QAAa,OAAO,C;MACpB,IAAI,SAAJ,C;QAAe,OAAO,E;MACTb,IAAI,SA AJ,C;QAAe,OAAO,C;MAGtB,OAA8B,iBAAtB,mDAAsB,EAAU,CAAV,C;K;IAaZ,6C;MAAA,uB;QAAU,2BAA oB,CAApB,EAAuB,CAAvB,EAA0B,iBAA1B,C;O;K;IAVhC,8B;MF7CI,IAAI,EEsDI,SAAU,OAAV,GAaiB,CfT DrB,CAAJ,C;QACI,cAda,qB;QAeb,MAAM,gCAAYB,OAAQ,WAAjC,C;;MEqDV,OAAO,eAAW,2BAAX,C;K;0F AIX,yB;MAAA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,O A5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MAPf,2B;QAOI,sBA AW,0BAAX,C;O;KAPJ,C;0FASA,yB;MAAA,oC;MAQe,gE;QAAA,uB;UAAU,iBAAsB,kB;UAAtB,eAAkC,gB;U AAIC,OA1Dd,UAAW,SAAQ,SA0DW,CA1DX,CAAR,EAAqB,SA0DC,CA1DD,CAArB,C;S;O;MAkDtB,uC;QA QI,sBAAW,sCAAX,C;O;KARJ,C;4GAUA,yB;MAAA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;U AAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EGb,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C; W;S;OA+EL,C;MAPf,2B;QAOI,sBAAW,oCAAX,C;O;KAPJ,C;8GASA,yB;MAAA,oC;MAUe,0E;QAAA,uB;UAA U,iBAAsB,kB;UAAtB,eAAkC,gB;UAAIC,OA/Ed,UAAW,SAAQ,SA+EW,CA/EX,CAAR,EAAqB,SA+EC,CA/ED ,CAArB,C;S;O;MAqEtB,uC;QAUI,sBAAW,gDAAX,C;O;KAVJ,C;kFAYA,yB;MAAA,sC;MAAA,oC;MAAA,oB AqE,yB;QA9Gf,8D;eA8Ge,yC;UAAA,uB;YACP,sBAAsB,WAAy,SAAQ,CAAR,EAAW,CAAX,C;YACIC,Q;YA AA,IAAI,oBAAmB,CAAvB,C;cAAA,OAA0B,e;;cAAqB,eAAsB,gB;cAArE,OAvgG,cAAc,SAuG8C,CAvG9C,C AAd,EAA2B,SAuGoC,CAvGpC,CAA3B,C;;YAsGH,W;W;S;OADO,C;MARf,sC;QAQI,sBAAW,kCAAX,C;O;K ARJ,C;oFAaA,yB;MAAA,oC;MAQe,0E;QAAA,uB;UACP,sBAAsB,WAAy,SAAQ,CAAR,EAAW,CAAX,C;UA CIC,Q;UAAA,IAAI,oBAAmB,CAAvB,C;YAAA,OAA0B,e;;YAAqB,iBAAsB,kB;YAAtB,eAAkC,gB;YAAjF,OA xGG,UAAW,SAAQ,SAwGyC,CAxGzC,CAAR,EAAqB,SAwG+B,CAxG/B,CAArB,C;;UAuGd,W;S;O;MATR,kD

;QAQI,sBAAW,8CAAX,C;O;KARJ,C;sGAaA,yB;MAAA,sC;MAAA,oC;MAAA,8BAQe,yB;QAxIf,8D;eAwIe,m  
D;UAAA,uB;YACP,sBAAsB,qBAAsB,SAAQ,CAAR,EAAW,CAAX,C;YAC5C,Q;YAAA,IAAI,oBAAmB,CAAv  
B,C;cAAA,OAA0B,e;;cAAqB,eAAsB,gB;cAArE,OAjIG,cAAc,SAiI8C,CAjI9C,CAAd,EAA2B,SAiIoC,CAjIpC,C  
AA3B,C;;YAgIH,W;W;S;OADO,C;MARf,sC;QAQI,sBAAW,4CAAX,C;O;KARJ,C;wGAaA,yB;MAAA,oC;MAQ  
e,8F;QAAA,uB;UACP,sBAAsB,qBAAsB,SAAQ,CAAR,EAAW,CAAX,C;UAC5C,Q;UAAA,IAAI,oBAAmB,CA  
AvB,C;YAAA,OAA0B,e;;YAAqB,iBAAsB,kB;YAAtB,eAAkC,gB;YAAjF,OAIIG,UAAW,SAAQ,SAkIyC,CAIz  
C,CAAR,EAAqB,SAkI+B,CAII/B,CAArB,C;;UAIId,W;S;O;MATR,kD;QAQI,sBAAW,wDAAX,C;O;KARJ,C;kG  
AcA,yB;MAAA,oC;MAOe,wE;QAAA,uB;UACP,sBAAsB,mBAAoB,SAAQ,CAAR,EAAW,CAAX,C;UAA1C,O  
ACI,oBAAmB,CAAvB,GAA0B,eAA1B,GAA+C,mBAAW,CAAX,EAAc,CAAd,C;S;O;MATvD,wC;QAOI,sBAA  
W,4CAAX,C;O;KAPJ,C;IAmBe,oD;MAAA,uB;QACP,sBAAsB,SAAU,SAAQ,CAAR,EAAW,CAAX,C;QAAhC,  
OACI,oBAAmB,CAAvB,GAA0B,eAA1B,GAA+C,kBAAW,SAAQ,CAAR,EAAW,CAAX,C;O;K;IATIE,uC;MAO  
I,sBAAW,kCAAX,C;K;IAYc,wE;MAAA,uB;QACV,sBAAsB,mBAAoB,SAAQ,CAAR,EAAW,CAAX,C;QAA1C,  
OACI,oBAAmB,CAAvB,GAA0B,eAA1B,GAA+C,kBAAW,SAAQ,CAAR,EAAW,CAAX,C;O;K;IATIE,+C;MAO  
I,sBAAc,4CAAd,C;K;IAaW,+C;MAAA,uB;QAEH,UAAM,CAAN,C;UADJ,OACe,C;aACX,c;UAFJ,OAEiB,E;aA  
Cb,c;UAHJ,OAGiB,C;;UAHjB,OAIY,kBAAW,SAAQ,CAAR,EAAW,CAAX,C;O;K;IAZ/B,gC;MAOI,sBAAW,6  
BAAX,C;K;4FASJ,yB;MAAA,4D;MAAA,wD;MAAA,mB;QAOqE,kBAAW,cAAx,C;O;KAPrE,C;IAGBe,8C;MA  
AA,uB;QAEH,UAAM,CAAN,C;UADJ,OACe,C;aACX,c;UAFJ,OAEiB,C;aACb,c;UAHJ,OAGiB,E;;UAHjB,OAI  
Y,kBAAW,SAAQ,CAAR,EAAW,CAAX,C;O;K;IAZ/B,+B;MAOI,sBAAW,4BAAX,C;K;0FASJ,yB;MAAA,4D;M  
AAA,sD;MAAA,mB;QAOoE,iBAAU,cAAV,C;O;KAPpE,C;IASA,wB;MAK4F,Q;MAA7B,OAA6B,4F;K;IAE5F,  
wB;MAK4F,Q;MAA7B,OAA6B,4F;K;IAE5F,gC;MAM+D,IAEJ,IAFI,EAGJ,M;MAFvD,kBAD2D,SAC3D,sB;QA  
DqD,OAC5B,SAAK,W;WAC9B,WAF2D,SAE3D,wC;QAFqD,OAEe,4F;WACvD,WAH2D,SAG3D,wC;QAHqD,  
OAGE,gG;;QAHF,OAI7C,uBAAmB,SAAnB,C;K;IAIuB,wC;MAAC,4B;K;2CACHC,gB;MAAwC,OAAA,eAAW,  
SAAQ,CAAR,EAAW,CAAX,C;K;4CACnD,Y;MACgC,sB;K;;IAGpC,kC;MAAA,sC;K;+CACI,gB;MAAoE,OAA  
E,iBAAF,CAAE,EAAU,CAAV,C;K;gDActE,Y;MAC8C,2C;K;;;IAHID,8C;MAAA,6C;QAAA,4B;;MAAA,sC;K;I  
AMA,kC;MAAA,sC;K;+CACI,gB;MAAoE,OAAE,iBAAF,CAAE,EAAU,CAAV,C;K;gDActE,Y;MAC8C,2C;K;;;  
IAHID,8C;MAAA,6C;QAAA,4B;;MAAA,sC;K;8EsKjTA,4B;MAUI,OAAK,iBAAL,SAAK,EAAU,KAAV,C;K;IC  
TT,iC;K;;;oDayDI,0C;MAiB+D,oB;QAAA,2C;aAjB/D,kG;K;;IAoBJ,uC;MAAA,e;MAAA,iB;MAAA,uB;K;IAA  
A,qC;MAAA,wC;O;MASI,4E;MAMA,8E;MAOA,4E;MAOA,kE;K;;IApBA,mD;MAAA,2B;MAAA,2C;K;;IAMA,  
oD;MAAA,2B;MAAA,4C;K;;IAOA,mD;MAAA,2B;MAAA,2C;K;;IAOA,8C;MAAA,2B;MAAA,sC;K;;IA7BJ,iC;  
MAAA,+K;K;;IAAA,sC;MAAA,a;AAAA,c;UAAA,gD;aAAA,e;UAAA,iD;aAAA,c;UAAA,gD;aAAA,S;UAAA,2C  
;;UAAA,oE;;K;;oFAqCA,mB;K;;;,,,,,,;lhImBiD,gD;MAAA,oB;QACzC,WAAW,sBAAmB,YAAF,CAAE,C  
AAnB,C;QACX,cAAM,IAAN,C;QADA,OAEA,IAAK,a;O;K;;;IAIhB,+B;K;;iFAUA,yB;MAAA,4B;MAAA,mC;  
QAMI,6BDgDQ,WChDkB,KDgDIB,CChDR,C;O;KANJ,C;2GAQA,yB;MAAA,4B;MDgDQ,kD;MChDR,uC;QAO  
I,6BDgDQ,WAAO,cChDW,SDgDX,CAAP,CChDR,C;O;KAPJ,C;+FAUA,yB;MAAA,kC;MAAA,mD;MAAA,yE;  
QASI,sC;QAAA,4C;O;MATJ,iGAWY,Y;QAAQ,2B;OAXpB,E;MAAA,0DAaQ,kB;QACI,wBAAW,MAAX,C;O;  
MAdZ,sF;MAAA,sC;QASI,0D;O;KATJ,C;IAiBA,gD;MAaI,4BAA0D,YAAzC,wCAA6B,UAA7B,CAAYC,CAA1  
D,EAAyE,yBAAzE,C;K;IAEJ,4D;MAcI,4BAAoE,YAAAnD,0CAA6B,QAA7B,EAAuC,UAAvC,CAAmD,CAApE,  
EAAmF,yBAAnF,C;K;IAEJ,+C;MAU6C,YAAzC,wCAA6B,UAA7B,CAAYC,CAtEzC,oBDgDQ,WCSBsD,kBDtB  
tD,CChDR,C;K;IAyEJ,2D;MAWuD,YAAAnD,0CAA6B,QAA7B,EAAuC,UAAvC,CAAmD,CAPFnD,oBDgDQ,Wc  
oCgE,kBDpChE,CChDR,C;K;IAuFJ,+C;MAYI,OAA6C,8BAAAtC,c;K;8EAZX,yB;MAAA,oE;MAAA,6E;MAYiD,  
gD;QAAA,oB;UACzC,WAAW,sBAAmB,YAAF,CAAE,CAAnB,C;UACX,cAAM,IAAN,C;UADA,OAEA,IAAK,  
a;S;O;MAfb,sC;QAYW,mBAAsC,8BAAAtC,6B;QAAP,OAAO,kD;O;KAZX,C;qGA0BI,yB;MAAA,2D;MAAA,mB  
;QACI,MAAM,6BAAoB,0BAApB,C;O;KADV,C;;MiIzIA,yC;;IAAA,uC;MAAA,2C;K;;;IAAA,mD;MAAA,kD;Q  
AAA,iC;;MAAA,2C;K;+EakBA,wB;K;oDAaA,e;MAK2C,IAAI,IAAJ,EAGK,M;MAL5C,IAAI,+CAAJ,C;QAEI,O  
AAW,GAAl,kBAAS,IAAK,IAAd,CAAR,GAA4B,cAAI,OAAJ,GAAl,iBAAQ,IAAR,CAAJ,yCAA5B,GAAyD,I;;  
MAGpE,OAAW,8CAA4B,GAAhC,GAAqC,8EAARc,GAAoD,I;K;yDAI/D,e;MAGI,IAAI,+CAAJ,C;QACI,OAA  
W,GAAl,kBAAS,IAAK,IAAd,CAAJ,IAA0B,GAAl,iBAAQ,IAAR,CAAJ,QAA9B,GAAyD,mCAAzD,GAAoF,I;;  
MAE/F,OAAW,8CAA4B,GAAhC,GAAqC,mCAArC,GAAgE,I;K;;;ICtChD,oD;MACf,cAAc,GAAl,kBAAS,OAA

Q,IAAjB,C;MACIB,IAAI,YAAY,mCAAhB,C;QADA,OACuC,O;;QAEnC,kBAaKB,oBAAQ,yCAAR,C;QACIB,I  
AAI,mBAAJ,C;UAJJ,OAI6B,oBAAgB,OAAhB,EAAYB,OAAzB,C;;UACrB,WAAW,OAAQ,kBAAS,yCAAT,C;U  
AL3B,OAMY,SAAS,mCAAb,GAAoC,oBAAgB,OAAhB,EAAYB,WAAzB,CAApC,GACI,oBAAgB,oBAAgB,IA  
AhB,EAAsB,OAAtB,CAAhB,EAAGD,WAAhD,C;;K;8CAdxB,mB;MAKI,OAAI,YAAY,mCAAhB,GAAuC,IAAv  
C,GACI,OAAQ,cAAK,IAAL,EAAW,4BAAX,C;K;;qDAiCZ,e;MAEyB,Q;MADrB,OACI,OAAA,IAAK,IAAL,  
EAAY,GAAZ,CAAJ,GAAqB,0EAArB,GAAoC,I;K;sDAExC,8B;MACI,iBAAU,OAAV,EAAMb,IAAnB,C;K;0D  
AEJ,e;MACI,OAAI,OAAA,IAAK,IAAL,EAAY,GAAZ,CAAJ,GAAqB,mCAArB,GAAgD,I;K;;IC1DP,8C;MAAC  
,wB;K;kFAAA,Y;MAAA,yB;K;;IAiCe,wD;MAEjE,kC;MAEA,4BAAqC,mDAAJ,GAaKd,OAAQ,qBAA1D,GAA  
0E,O;K;4DAE3G,mB;MAA6C,+BAAS,OAAI,C;K;6DAC7C,e;MAA8C,eAAQ,IAAR,IAAgB,8BAAE,G;K;;IAGjF  
,+C;MAW2C,IAAI,IAAJ,EAGV,M;MAL7B,IAAI,+CAAJ,C;QAEI,OAAW,GAAl,kBAAS,SAAK,IAAd,CAAR,G  
AA4B,cAAI,OAAJ,GAAl,iBAAQ,SAAR,CAAJ,yCAA5B,GAAYD,I;;MAGpE,OAAW,SAAK,IAAL,KAAa,GAAj  
B,GAAsB,mFAAtB,GAAqC,I;K;IAGhD,6C;MAUI,IAAI,+CAAJ,C;QACI,OAAW,GAAl,kBAAS,SAAK,IAAd,C  
AAJ,IAA0B,GAAl,iBAAQ,SAAR,CAAJ,QAA9B,GAAYD,mCAAzD,GAAoF,S;;MAE/F,OAAW,SAAK,IAAL,K  
AAa,GAAjB,GAAsB,mCAAtB,GAAd,S;K;IAG5D,iC;MAAA,qC;MAKI,4B;K;oDACA,Y;MAAiC,0C;K;kDAEj  
C,e;MAAYD,W;K;mDACzD,8B;MAA4E,c;K;mDAC5E,mB;MAAwE,c;K;uDACxE,e;MAA8D,W;K;+CAC9D,Y;  
MAAsC,Q;K;+CACtC,Y;MAAYC,8B;K;;IAb7C,6C;MAAA,4C;QAAA,2B;;MAAA,qC;K;IAqB8B,wC;MAC1B,k  
B;MACA,wB;K;4CAGA,e;MAGQ,Q;MAFJ,UAAU,I;MACV,OAAO,IAAP,C;QACI,YAAA,GAAl,UAAJ,aAAY,  
GAAZ,W;UAAwB,W;;QACxB,WAAW,GAAl,O;QACf,IAAI,oCAAJ,C;UACI,MAAM,I;;UAEN,OAAO,iBAAK,  
GAAL,C;;K;6CAKnB,8B;MACI,iBAAU,WAAK,cAAK,OAAI,EAAC,SAAd,CAAF,EAAYC,cAAzC,C;K;iDAEJ,e  
;UAGW,I;MAFP,+BAAQ,GAAR,U;QAAoB,OAAO,W;;MAC3B,cAAc,WAAK,kBAAS,GAAT,C;MAEf,gBAAY,  
WAAZ,C;QAAoB,W;WACpB,gBAAY,mCAAZ,C;QAAqC,qB;;QAC7B,2BAAgB,OAAhB,EAAYB,cAAzB,C;MA  
HZ,W;K;uCAOJ,Y;MAIc,IAAI,IAAJ,Q;MAHV,UAAU,I;MACV,WAAW,C;MACX,OAAO,IAAP,C;QACU,uBA  
Al,OAAJ,GAAl,OAAJ,gC;QAAA,mB;UAAgC,OAAO,I;;QAA7C,MAAM,M;QACN,mB;;K;2CAIR,mB;MACI,+  
BAAl,OAAQ,IAAZ,GAAoB,OAApB,C;K;8CAEJ,mB;MAQ4B,Q;MAPxB,UAAU,O;MACV,OAAO,IAAP,C;QA  
CI,IAAI,CAAC,gBAAS,GAAl,UAAb,CAAL,C;UAA4B,OAAO,K;QACnC,WAAW,GAAl,O;QACf,IAAI,oCAAJ,  
C;UACI,MAAM,I;;UAEN,OAAO,gBAAS,0EAAT,C;;K;uCAKnB,iB;MACI,gBAAS,KAAT,KAaKB,yCAA4B,K  
AAM,SAAN,KAAGB,aAA5C,IAAsD,KAAM,eAAY,IAAZ,CAA9E,C;K;yCAEJ,Y;MAA+B,OAAK,SAAL,WAA  
K,CAAL,GAA0B,SAAR,cAAQ,CAA1B,I;K;IAGZ,uD;MACX,OAAI,GzJyHoC,YAAU,CyJzHID,GAAMb,OAAQ  
,WAA3B,GAA6C,GAAF,UAAQ,O;K;yCAF3D,Y;MACI,aAAM,kBAAK,EAAL,EAAS,+BAAT,CAAN,GAEL,G;  
K;IAMO,8E;MAAA,6B;QAAyB,Q;QAAT,iBAAS,sBAAT,EAAS,8BAAT,UAAoB,O;QAAQ,W;O;K;+CAJ3D,Y;  
MAOsB,Q;MANIB,QAAQ,a;MACR,eAAe,gBAA+B,CAA/B,O;MACf,gBAAY,CAAZ,C;MACA,kBAAK,kBAAL  
,EAAW,oDAAX,C;M5KtFJ,IAAI,E4KuFM,YAAS,C5KvFf,CAAJ,C;QACI,cAdW,e;QAEX,MAAM,6BAAsB,OA  
AQ,WAA9B,C;;M4KuFN,OAAO,+BAAW,qDAAX,C;K;IAGa,8C;MACpB,kD;MADqB,wB;K;IACrB,gD;MAAA,  
oD;MACI,4B;K;;IADJ,4D;MAAA,2D;QAAA,0C;;MAAA,oD;K;yDAIA,Y;MAA0C,gBAAT,a;M3L09YrB,Q;MA  
DhB,kB2Lz9YmD,mC;M3L09YnD,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,cAAwB,yBAAa,OA  
Ab,C;;M2L19YT,O3L29Y9B,W;K;;I4L7oZX,oE;MA4BI,MAAM,wBAAoB,sEAAPB,C;K;8GA5BV,yB;MAAA,2  
D;MAAA,sC;QA4BI,MAAM,6BAAoB,sEAAPB,C;O;KA5BV,C;IA0CoC,mC;MAAQ,4D;K;IAE5C,4C;MAAA,e;  
MAAA,iB;MAAA,uB;K;IAAA,0C;MAAA,6C;O;MAK0C,oG;MAAQB,gF;MAAW,4E;K;;IAAhC,+D;MAAA,gC;  
MAAA,uD;K;;IAAQB,qD;MAAA,gC;MAAA,6C;K;;IAAW,mD;MAAA,gC;MAAA,2C;K;;IAL1E,sC;MAAA,sJ;K  
;;IAAA,2C;MAAA,a;aAAA,qB;UAAA,4D;aAAA,W;UAAA,kD;aAAA,S;UAAA,gD;;UAAA,qF;;K;;6ECnDA,yB;  
MAAA,0B;MAAA,mC;QAGsD,OAAiC,OAA3B,SAAL,GAAuB,KAAS,C;O;KAHvF,C;2EAKA,yB;MAAA,0B;M  
AAA,mC;QAGqD,OAAgC,OAA1B,SAAL,GAAsB,KAAS,C;O;KAHrF,C;6EAKA,yB;MAAA,0B;MAAA,mC;QA  
GsD,OAAiC,OAA3B,SAAL,GAAuB,KAAS,C;O;KAHvF,C;6EAKA,yB;MAAA,0B;MAAA,4B;QAGqC,OAAqB,  
OAAp,CAAR,SAAE,C;O;KAH1D,C;+EAMA,yB;MAAA,4B;MAAA,mC;QAGyD,OAAiC,QAA3B,SAAL,GAAu  
B,KAAS,C;O;KAH1F,C;6EAKA,yB;MAAA,4B;MAAA,mC;QAGwD,OAAgC,QAA1B,SAAL,GAAsB,KAAS,C;  
O;KAHxF,C;+EAKA,yB;MAAA,4B;MAAA,mC;QAGyD,OAAiC,QAA3B,SAAL,GAAuB,KAAS,C;O;KAH1F,C;  
+EAKA,yB;MAAA,4B;MAAA,4B;QAGuC,OAAqB,QAAP,CAAR,SAAE,C;O;KAH5D,C;ICpCA,qC;K;;ICAA,m  
B;K;;IAOA,iB;K;;IAOA,2C;K;;IAOA,wB;K;;IAQA,0B;K;;IAOA,sB;K;;IAOA,4B;K;;IAOA,6C;K;;IA+BuC,wE;M

AEnC,uB;QAAA,UAAaB,E;MACtB,qB;QAAA,8B;MACA,2B;QAAA,qE;MACA,yB;QAAA,YAAqB,E;MAJrB,s  
B;MACA,sB;MACA,kB;MACA,8B;MACA,0B;K;;IAGJ,iD;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,+C;MAAA,  
kD;O;MAKI,wG;MACA,wG;MACA,8F;K;;IAFA,iE;MAAA,qC;MAAA,yD;K;;IACA,iE;MAAA,qC;MAAA,yD;K  
;;IACA,4D;MAAA,qC;MAAA,oD;K;;IAPJ,2C;MAAA,6K;K;;IAAA,gD;MAAA,a;aAAA,kB;UAAA,8D;aAAA,kB  
;UAAA,8D;aAAA,a;UAAA,yD;;UAAA,6E;;K;;IAUA,wB;K;;ICnGA,mB;MAEI,UAAU,IAAI,CAAJ,I;MACV,OA  
AW,OAAO,CAAX,GAAC,GAAd,GAAuB,MAAM,CAAN,I;K;IAGIC,qB;MACI,UAAU,SAAI,CAAJ,C;MACV,O  
AAW,kBAAO,CAAX,GAAC,GAAd,GAAuB,QAAM,CAAN,C;K;IAGIC,mC;MAEI,OAAO,IAAI,IAAI,CAAJ,EA  
AO,CAAP,IAAY,IAAI,CAAJ,EAAO,CAAP,CAAZ,IAAJ,EAA2B,CAA3B,C;K;IAGX,qC;MACI,OAAO,MAAI,M  
AAI,CAAJ,EAAO,CAAP,WAAY,MAAI,CAAJ,EAAO,CAAP,CAAZ,CAAJ,EAA2B,CAA3B,C;K;IAGX,qD;MAK  
BI,WAAO,CAAP,C;QAD2E,OAC3D,SAAS,GAAb,GAakB,GAAIB,GAA2B,MAAM,iBAaIB,GAAjB,EAAaB,K  
AAiB,EAA6B,IAA7B,CAAN,I;WACvC,WAAO,CAAP,C;QAF2E,OAE3D,SAAS,GAAb,GAakB,GAAIB,GAA2  
B,MAAM,iBAaIB,KAAjB,EAAwB,GAAxB,EAA6B,CAAC,IAAD,IAA7B,CAAN,I;;QAC/B,MAAa,gCAAYB,eA  
AzB,C;K;IAGzB,uD;MAKBI,sBAAO,CAAP,C;QAD+E,OAC/D,sBAAS,GAAT,MAAJ,GAakB,GAAIB,GAA2B,a  
AAM,mBAaIB,GAAjB,EAAaB,KAAiB,EAA6B,IAA7B,CAAN,C;WACvC,sBAAO,CAAP,C;QAF+E,OAE/D,sB  
AAS,GAAT,MAAJ,GAakB,GAAIB,GAA2B,QAAM,mBAaIB,KAAjB,EAAwB,GAAxB,EAA8B,IAAD,aAA7B,  
CAAN,C;;QAC/B,MAAa,gCAAYB,eAAzB,C;K;IC7DzB,qB;MAAA,yB;K;0CAII,Y;MAO6D,uB;K;2HAE7D,yB;  
MAAA,+D;MAAA,kC;MAAA,0F;MAAA,6F;MAAA,4E;QAUI,wC;QAAS,2C;O;MAVb,mEAWQ,wC;QAA6E,sB  
AAS,QAAT,EAAmB,QAAnB,EAA6B,QAA7B,C;O;MAXrF,oG;MAAA,yC;QAUI,wDAA+B,YAA/B,C;O;KAVJ,  
C;uHAcA,yB;MAAA,+D;MAAA,kC;MAAA,wF;MAAA,yF;MAAA,0E;QAcI,wC;QAAS,2C;O;MADB,kEAeQ,wC  
;QAAuF,6BAAS,QAAT,EAAmB,QAAnB,EAA6B,QAA7B,C;O;MAfF,kG;MAAA,yC;QAcI,sDAA+B,YAA/B,C;  
O;KAdJ,C;;IA3BJ,iC;MAAA,gC;QAAA,e;;MAAA,yB;K;IAgDiC,sB;MAC7B,eAAwB,I;K;4CAExB,6B;MACW,  
Q;MAAA,mB;MAAA,iB;QAAS,MAAM,6BAASB,cAAy,QAAS,aAArB,uCAAtB,C;;MAAtB,OAAO,I;K;4CAGX,  
oC;MACI,eAAa,K;K;;;kDC9CjB,6B;;K;;;iEA+CA,6B;;K;;ICrDuC,0C;MACvC,uBAAoB,Y;K;wDAEPB,wC;  
MAM6F,W;K;uDAE7F,wC;K;ODAMA,6B;MACI,OAAO,oB;K;ODAGX,oC;MACI,eAAe,IAAK,gB;MACpB,IAAI  
,CAAC,0BAAa,QAAb,EAAuB,QAAvB,EAAiC,KAAjC,CAAL,C;QACI,M;;MAEJ,uBAAa,K;MACb,yBAAY,QA  
AZ,EAAaB,QAAtB,EAAgC,KAAhC,C;K;;4EC9BR,wC;MAqBI,OAAO,e;K;4EAGX,+C;MAuBI,cAAI,KAAJ,C;K  
;4EAIJ,wC;MAmBI,OAAO,cAAI,OAAJ,C;K;4EAGX,+C;MAqBI,cAAI,OAAJ,EAAa,KAAb,C;K;IC/FJ,kB;MA6P  
I,4B;K;+BAtoA,Y;MAOic,6BAAS,EAAT,C;K;uCAEjC,iB;MAW2C,4BAAQ,CAAR,EAAW,KAAx,C;K;uCAE3  
C,uB;MAakB,Q;MAHd,iBAaIB,IAAjB,EAAuB,KAAvB,C;MACA,QAAQ,QAAQ,IAAR,I;MACR,IAAI,IAAI,CA  
AJ,IAAS,MAAK,WAAIB,C;QACc,IAAI,MAAM,CAAC,CAAD,IAAN,OAAy,CAAhB,C;UACN,eAAe,SAAS,CA  
AT,C;UACf,6BAAS,QAAT,C;;UAEA,K;;YAEI,WAAW,cAAU,KAAK,C;YACIB,IAAI,OAAO,CAAP,I;;UACC,g  
BAAO,CAAP,IAAY,CAAZ,GAAgB,CAAhB,SAaQb,CAArB,C;UACT,Q;;QATJ,c;QAWA,OAAO,OAAO,GAAP  
,I;;QAEp,OAAO,IAAP,C;UACI,YAAU,c;UACV,IAAW,IAAP,qBAakB,KAAiB,C;YAA6B,OAAO,K;;K;gCAKh  
D,Y;MAOmC,OAAU,oBAAV,cAAU,CAAS,WAAI,EAAJ,CAAnB,yBAA6B,cAA7B,E;K;wCAEnC,iB;MAW8C,i  
CAAY,KAAZ,C;K;wCAE9C,uB;MAiBkB,Q;MAPd,mBAaIB,IAAjB,EAAuB,KAAvB,C;MACA,QAAQ,eAAQ,I  
AAR,C;MACR,IAAI,eAAI,CAAR,C;QACI,O;QACA,IAAI,aAAO,CAAD,aAAN,GAAY,CAAZ,CAAJ,C;UACI,W  
AAW,CAAE,Q;UACb,YAAa,qBAAO,EAAP,CAAW,Q;UAEpB,aAAQ,CAAR,C;YACI,eAAe,SAAS,IAAT,C;YA  
Ef,OAAmB,oBAAnB,sBAAS,QAAT,CAAmB,CAAnB,iB;IBAEJ,cAAS,CAAT,C;YAEI,OAAU,oBAAV,cAAU,C  
AAV,iB;;YAEA,iBAaE,SAAS,KAAT,C;YACf,OAAmB,oBAAnB,sBAAS,UAAT,CAAmB,CAAS,WAAI,EAAJ,C  
AA5B,KAAiD,oBAAV,cAAU,CAAV,iBAAvC,C;;UAXR,U;;UAEa,K;;YAEI,WAAW,eAAW,oBAAK,CAAL,C;  
YACTb,IAAI,YAAO,CAAP,C;;UACC,sBAAO,CAAP,MAAY,+BAAI,CAAJ,EAAZ,eAAqB,CAArB,C;UACT,MA  
AM,C;;QAEV,OAAO,SAAO,GAAP,C;;QAEp,OAAO,IAAP,C;UACI,YAAU,e;UACV,IAAW,IAAP,0CAakB,KA  
AIB,CAAJ,C;YAA6B,OAAO,K;;K;mCAKhD,Y;MAKyC,6BAAS,CAAT,MAAE,C;K;kCAExD,Y;MAKuC,uBAA  
gB,sBAAS,EAAT,CAAhB,EAA8B,sBAAS,EAAT,CAA9B,C;K;0CAEvC,iB;MASoD,+BAAW,GAAX,EAAgB,K  
AAhB,C;K;0CAEpD,uB;MAcY,Q;MAFR,mBAaIB,IAAjB,EAAuB,KAAvB,C;MACA,WAAW,QAAQ,I;MACX,I  
AAS,WAAI,IAAK,CAAL,IAA0B,SAAL,IAAK,CAA1B,IAA8C,SAAN,KAAM,CAAD,C;QACJ,SAAS,qBAAgB  
,QAAQ,CAAR,GAAY,OAAO,CAAnC,C;QACT,cAAO,EAAP,GAAY,E;;QAEZ,cAAO,oBAaE,I;;MAJ1B,Y;MA  
MA,OAAW,KAAK,KAAT,GAASB,SAAN,KAAM,CAAtB,GAAS,C;K;iCAGjD,Y;MAKqC,6BAAS,EAAT,IAA

0B,Q;K;IAWK,oF;MAAA,mB;QAAE,uBAAa,iBAAb,sBAAqC,eAArC,+BAAqE,aAAM,OAA3E,M;O;K;iDATtE,  
qC;MtlJLA,IAAI,EsL0LqB,CAAb,8BAAgB,KAAM,Otl1L9B,GsL0LiD,CAAX,0BAAc,KAAM,Otl1L1D,GsL0L  
sC,KtL1LtC,CAAJ,C;QACI,csLyLgE,kDtLzLID,E;QACd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MAFV,IAAI,EsL2  
LQ,aAAa,Otl3LrB,CAAJ,C;QACI,gBsL0LgC,mF;QtLzLhC,MAAM,gCAAYB,SAAQ,WAAjC,C;;MsL2LN,YAA  
Y,CAAC,UAAU,SAAV,IAAD,IAAwB,CAAxB,I;MAEZ,mBAAe,SAaf,C;MnLzEJ,iBAAc,CAAd,Uml0EW,KnL  
lEX,U;QmL2EQ,QAAQ,c;QACR,MAAM,UAAAN,IAAoB,OAAf,CAAE,C;QACpB,MAAM,aAAW,CAAX,IAAN,  
IAAgC,OAAV,CAAE,KAAK,CAAG,C;QACHC,MAAM,aAAW,CAAX,IAAN,IAAiC,OAAx,CAAE,KAAK,EAA  
I,C;QACjC,MAAM,aAAW,CAAX,IAAN,IAAiC,OAAx,CAAE,KAAK,EAAI,C;QACjC,0BAAY,CAAZ,I;;MAGJ,  
gBAAgB,UAAU,UAAV,I;MACHB,SAAS,sBAAS,YAAy,CAAZ,IAAT,C;MACT,aAAU,CAAV,MAAkB,SAAIB,  
M;QACI,MAAM,aAAW,CAAX,IAAN,IAAqC,OAAf,EAAG,MAAK,IAAI,CAAJ,IAAL,CAAY,C;;MAGzC,OAA  
O,K;K;yCACX,uD;MAvB4C,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,KAAM,O;aArRF,0H;K;yCAiCA,iB;  
MAOyD,8BAAU,KAAV,EAAiB,CAAjB,EAAoB,KAAM,OAA1B,C;K;yCAEzD,gB;MAKkD,8BAAU,cAAU,IAA  
V,CAAV,C;K;IAGiD,0B;MAAA,8B;MAO2B,iB;MACvB,uBAAoC,uB;K;IAEpC,qC;MAAA,yC;MACI,4B;K;wD  
AEA,Y;MAAiC,mC;K;;IAHrC,iD;MAAA,gD;QAAA,+B;;MAAA,yC;K;8CAMA,Y;MAAkC,8C;K;gDAEIC,oB;  
MAA4C,OAAA,oBAAc,kBAAS,QAAT,C;K;uCAC1D,Y;MAA8B,OAAA,oBAAc,U;K;+CAC5C,iB;MAAwC,OA  
AA,oBAAc,iBAAQ,KAAR,C;K;+CACtD,uB;MAAmD,OAAA,oBAAc,iBAAQ,IAAR,EAAC,KAAd,C;K;wCAEjE,  
Y;MAAgC,OAAA,oBAAc,W;K;gDAC9C,iB;MAA2C,OAAA,oBAAc,kBAAS,KAAT,C;K;gDACzD,uB;MAAuD,  
OAAA,oBAAc,kBAAS,IAAT,EAae,KAAf,C;K;2CAErE,Y;MAAsC,OAAA,oBAAc,c;K;0CAEpD,Y;MAAoC,OA  
AA,oBAAc,a;K;kDACID,iB;MAAiD,OAAA,oBAAc,oBAAW,KAAX,C;K;kDAC/D,uB;MAA+D,OAAA,oBAAc,  
oBAAW,IAAX,EAAiB,KAAjB,C;K;yCAE7E,Y;MAAkC,OAAA,oBAAc,Y;K;iDAEhD,iB;MAAsD,OAAA,oBAA  
c,mBAAU,KAAV,C;K;iDACpE,gB;MAA+C,OAAA,oBAAc,mBAAU,IAAV,C;K;yDAC7D,qC;MACI,OAAA,oB  
AAc,mBAAU,KAAV,EAAiB,SAAjB,EAA4B,OAA5B,C;K;;IAtCtB,sC;MAAA,qC;QAAA,oB;;MAAA,8B;K;;IA  
0CJ,wB;MAAuC,yBAAa,IAAb,EAAMB,IAAK,IAAI,EAA5B,C;K;IAEvC,wB;MAAwC,yBAAa,IAAK,QAAIB,EA  
A2B,IAAK,YAAI,EAAJ,CAAQ,QAAX,C;K;IAGxC,mC;MAUI,IAAA,KAAM,UAAAN,C;QAAMB,MAAM,gCA  
AyB,uCAAoC,KAA7D,C;WACzB,IAAA,KAAM,KAAN,GAAa,UAAb,C;QAF8C,OAeHb,0BAAQ,KAAM,MAA  
d,EAAqB,KAAM,KAAN,GAAa,CAAb,IAArB,C;WAC9B,IAAA,KAAM,MAAN,GAAc,WAAAd,C;QAH8C,OAGf,  
0BAAQ,KAAM,MAAN,GAAc,CAAd,IAAR,EAAYB,KAAM,KAA/B,IAAuC,CAAV,C;QAEhE,OAIc,mB;K;IAG  
Z,oC;MAUI,IAAA,KAAM,UAAAN,C;QAAMB,MAAM,gCAAYB,uCAAoC,KAA7D,C;WACzB,IAAA,KAAM,KA  
AN,+C;QAFiD,OAeIB,2BAAS,KAAM,MAAf,EAAsB,KAAM,KAAN,yBAAa,CAAb,EAATB,C;WAC/B,IAAA,K  
AAM,MAAN,+C;QAHID,OAGjB,2BAAS,KAAM,MAAN,8BAAc,CAAd,EAAT,EAA0B,KAAM,KAAhC,0BAA  
wC,CAAX,C;QAHIB,OAIzC,oB;K;IAOZ,yB;MAAYC,YnFrTkB,YmFqTb,KnFrTa,CmFqTIB,I;K;IAEzC,4C;MA  
EI,OAAA,SAAK,KAAK,EAAL,GAAU,QAAf,GAAyC,CAAX,CAAC,QAAD,IAAW,KAAI,E;K;IAEjD,uC;Mtl  
VI,IAAI,EsLsVuD,QAAQ,ItLtV/D,CAAJ,C;QACI,csLqVuE,+B;QtLpVvE,MAAM,gCAAYB,OAAQ,WAAjC,C;;K  
;IsLqVd,yC;MtlvVI,IAAI,EsLuVyD,sBAAQ,IAAR,KtLvVzD,CAAJ,C;QACI,csLsVyE,+B;QtLrVzE,MAAM,gC  
AAyB,OAAQ,WAAjC,C;;K;IsLsVd,yC;MtlXVI,IAAI,EsLwV6D,QAAQ,ItLxVrE,CAAJ,C;QACI,csLuV6E,+B;Qt  
LtV7E,MAAM,gCAAYB,OAAQ,WAAjC,C;;K;IsLwVd,yC;MAAYD,oCAA0B,IAA1B,qBAAiC,KAAjC,kB;K;ICr  
XzD,6B;MAOqC,OnMmYE,SmMnYF,mBnMmYE,C;K;ImMjYvC,sC;MASgD,6BAAS,WAAT,EAAa,KAAb,C;K  
;IAEhD,4C;MAUI,qBAAqB,IAArB,EAA2B,KAA3B,C;MAEA,iBAAiB,InMqQgB,KmMrQhB,GAAiB,W;MACIC  
kBAAkB,KnMoQe,KmMpQf,GAAkB,W;MAEpC,mBAAmB,0BAAQ,UAAR,EAAoB,WAApB,IAAqC,W;MACx  
D,OnMsWmC,SmMtW5B,YnMsW4B,C;K;ImMnWvC,sC;MAWI,IAAA,KAAM,UAAAN,C;QAAMB,MAAM,gCA  
AyB,uCAAoC,KAA7D,C;;QACzB,InMGkE,YmMHIE,KAAM,KnMG6E,KAAjB,EmMHRd,4BAAK,UnMG6E,K  
AA7B,CmMHIE,K;UAFiD,OAeIB,sBAAS,KAAM,MAAf,EnMqBsB,SmMrBA,KAAM,KnMqBI,KAAK,GAAW,  
CmMrBb,WnMqBa,MAAX,IAAf,CmMrBtB,C;;UAC/B,InMEkE,YmMFIE,KAAM,MnME6E,KAAjB,EmMFPd,4  
BAAK,UnME4E,KAA7B,CmMFIE,K;YAHID,OnMuBI,SmMpBrB,sBnMiCsB,SmMjCb,KAAM,MnMiCiB,KAA  
K,GAAy,CmMjC1B,WnMiC0B,MAAZ,IAAf,CmMjCtB,EAA2B,KAAM,KAAjC,CnMoB+B,KAAK,GAAW,Cm  
MpBN,WnMoBM,MAAX,IAAf,C;;YmMvBJ,OAIzC,mB;;K;IAGZ,8B;MAOuC,OIL0VG,UkL1VH,oBIL0VG,C;K  
;IkLxV1C,uC;MASmD,8BAAU,2BAAV,EAAe,KAAf,C;K;IAEnD,6C;MAUI,sBAAsB,IAATB,EAA4B,KAA5B,C;  
MAEA,iBAAiB,IILwNkB,KkLxNIB,8B;MACjB,kBAAkB,KILuNiB,KkLvNjB,8B;MAEIB,mBAAmB,2BAAS,UA

AT,EAAqB,WAArB,+B;MACnB,OIL6TsC,UkL7T/B,YIL6T+B,C;K;IkL1T1C,uC;MAWI,IAAA,KAAM,UAAN,C;  
;QAAMb,MAAM,gCAAYb,uCAAoC,KAA7D,C;;QACzB,IIL7CmE,akL6CnE,KAAM,KIL7C+E,KAAIB,EkL6Ct  
D,6BAAM,UIL7C8E,KAA9B,CkL6CnE,K;UAFoD,OAEPb,uBAAU,KAAM,MAAhB,EILhCuB,UkLgCA,KAAM,  
KILhCK,KAAK,KAAW,CjBsQ7C,UAAW,oBAAL,CmMtOyB,WnMsOzB,MAAK,CAAL,iBAAN,CiBtQ6C,MAA  
X,CAAhB,CkLgCvB,C;;UACHC,IIL9CmE,akL8CnE,KAAM,MIL9C+E,KAAIB,EkL8CrD,6BAAM,UIL9C6E,KA  
A9B,CkL8CnE,K;YAHoD,OIL9BG,UkLiCtB,uBILpBuB,UkLoBb,KAAM,MILpBkB,KAAK,UAAy,CjByP/C,UA  
AW,oBAAL,CmMrOc,WnMqOd,MAAK,CAAL,iBAAN,CiBzP+C,MAAZ,CAAhB,CkLoBvB,EAA4B,KAAM,K  
AAIC,CILjCiC,KAAK,KAAW,CjBsQ7C,UAAW,oBAAL,CmMrOgC,WnMqOhC,MAAK,CAAL,iBAAN,CiBtQ6  
C,MAAX,CAAhB,C;;YkL8BH,OAI5C,oB;;;K;IAGZ,sC;MAQI,4BAAU,K/Jg/FH,Q+Jh/FP,C;MACA,OAAO,K;K;I  
AGX,uC;MAKsD,O/J+iG3C,e+J/iG2C,4BAAU,IAAV,C/J+iG3C,C;K;I+J7iGX,4D;MAOGD,yB;QAAA,YAAiB,C;  
MAAG,uB;QAAA,UAAe,KAAM,K;MACrF,4BAAU,K/J69FH,Q+J79FP,EAA+B,SAA/B,EAA0C,OAA1C,C;MA  
CA,OAAO,K;K;IAIX,2C;MvLrHI,IAAI,EZ2B8D,YmM0FD,KnM1FkB,KAAjB,EmM0FO,InM1FsB,KAA7B,Cm  
M0FD,IvLrH7D,CAAJ,C;QACI,cuLoH6E,+B;QvLnH7E,MAAM,gCAAYB,OAAQ,WAAjC,C;;K;IuLoHd,4C;MvL  
tHI,IAAI,EKMc+D,akLmFC,KILnFiB,KAAIB,EkLmFS,IILnFqB,KAA9B,CkLmFC,IvLrHhE,CAAJ,C;QACI,cuLq  
HgF,+B;QvLpHhF,MAAM,gCAAYB,OAAQ,WAAjC,C;;K;IwLpBc,6C;MAcxB,oC;MA/BA,iB;MANA,Y;MAC  
A,Y;MACA,Y;MACA,Y;MACA,Y;MACA,sB;MxLYA,IAAI,EwLLQ,CAAC,WAAK,QAAL,GAAU,QAAY,GAA  
e,QAaf,GAAoB,QAARb,MAA2B,CxLKnC,CAAJ,C;QACI,cwLNwC,wD;QxLOxC,MAAM,gCAAYB,OAAQ,WA  
AjC,C;;MGoHV,iBAAc,CAAd,UqLxHW,ErLwHX,U;QqLxHiB,c;;K;qCAGjB,Y;MAGI,QAAQ,Q;MACR,IAAI,I  
AAO,MAAO,C;MACIB,WAAI,Q;MACJ,WAAI,Q;MACJ,WAAI,Q;MACJ,SAAS,Q;MACT,WAAI,E;MACJ,IAA  
K,IAAO,KAAM,CAAd,GAA8B,MAAO,C;MACzC,WAAI,C;MACJ,gCAAU,MAAV,I;MACA,O  
AAO,IAAI,aAAJ,I;K;8CAGX,oB;MACI,OAAU,cAAV,cAAU,EAAC,QAAd,C;K;IAEd,kC;MAAA,sC;MACI,4B;K  
;;IADJ,8C;MAAA,6C;QAAA,4B;;MAAA,sC;K;;IA7BA,gD;MAAA,sD;MACQ,yBAAK,KAAL,EAAy,KAaz,E  
AAmB,CAAnB,EAASB,CAAtB,EAA+B,CAAN,KAazB,EAAuC,SAAU,EAAX,GAAoB,UAAW,CAARe,C;MAD  
R,Y;K;I/LbKB,wC;MA8BIB,iC;MA9BsD,2BAAGb,KAAhB,EAAuB,YAAvB,EAAqC,CAARc,C;K;kFAC7B,Y;M  
AAQ,8B;K;yFACD,Y;MAAQ,6B;K;yFAKR,Y;MAC5B,IAAI,cAAQ,sCAAK,UAAjB,C;QOyHyC,MAAM,6BPzH  
b,6EOyH2C,WAA9B,C;;MPxH/C,OAAO,+BAAO,CAAP,E;K;2CAGX,iB;MAA8C,qBAAS,KAAT,IAAkB,SAAS,  
S;K;kCAEzE,Y;MAKkC,oBAAQ,S;K;iCAE1C,iB;MACI,2CAAuB,kBAaA,KAAM,UAAAnB,KACvB,eAAS,KAA  
M,MAAf,IAAwB,cAAQ,KAAM,KADf,CAAvB,C;K;mCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAaf,GAAwB,OAAK  
,UsBUS,ItBVd,UAAkB,SsBUJ,ItBVd,K;K;mCAE5B,Y;MAAkC,2BAAE,UAAF,+BAAU,SAAV,C;K;IAEIC,+B;  
MAAA,mC;MACI,aAC8B,cAAY,OAAF,CAAE,CAAZ,EAAwB,OAAF,CAAE,CAAxB,C;K;;IAFIC,2C;MAAA,0  
C;QAAA,yB;;MAAA,mC;K;;IAUiB,uC;MA8BjB,gC;MA9BmD,0BAAE,KAAf,EAASB,YAAtB,EAAoC,CAAPc,  
C;K;iFAC3B,Y;MAAQ,iB;K;wFACD,Y;MAAQ,gB;K;wFAKR,Y;MAC3B,IAAI,cAAQ,UAAZ,C;QOiFyC,MAA  
M,6BPjFd,6EOiF4C,WAA9B,C;;MPHF/C,OAAO,YAAO,CAAP,I;K;0CAGX,iB;MAA6C,qBAAS,KAAT,IAAkB,  
SAAS,S;K;iCAExE,Y;MAKkC,oBAAQ,S;K;gCAE1C,iB;MACI,0CAASB,kBAaA,KAAM,UAAAnB,KACtB,eAAS,  
KAAM,MAAf,IAAwB,cAAQ,KAAM,KADhB,CAAtB,C;K;kCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAaf,GAAwB,  
MAAK,UAAAL,QAAa,SAAb,I;K;kCAE5B,Y;MAAkC,OAAE,UAAF,qBAAU,S;K;IAE5C,8B;MAAA,kC;MACI,a  
AC6B,aAAS,CAAT,EAAy,CAAZ,C;K;;IAFjC,0C;MAAA,yC;QAAA,wB;;MAAA,kC;K;;IAUkB,wC;MA8BIB,i  
C;MA9BsD,2BAAGb,KAAhB,EAAuB,YAAvB,K;K;kFAC7B,Y;MAAQ,iB;K;yFACD,Y;MAAQ,gB;K;yFAKR,Y;  
MAC5B,IAAI,2CAAJ,C;QOyCyC,MAAM,6BPzCb,6EOyC2C,WAA9B,C;;MPxC/C,OAAO,kCAAo,CAAP,E;K;2  
CAGX,iB;MAA8C,kCAAS,KAAT,UAAkB,sBAAS,SAAT,M;K;kCAEhE,Y;MAKkC,kCAAQ,SAAR,K;K;iCAEIC  
,iB;MACI,2CAAuB,kBAaA,KAAM,UAAAnB,KACvB,mBAAS,KAAM,MAAf,KAawB,kBAAQ,KAAM,KAAd,C  
ADD,CAAvB,C;K;mCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAaf,GAAwB,iCAAM,eAAW,8BAAW,EAAX,CAAX,  
CAAN,MAAoC,cAAU,6BAAU,EAAY,CAAV,CAAPc,CAA8D,Q;K;mCAE1F,Y;MAAkC,OAAE,UAAF,qBAAU  
,SAAV,W;K;IAEIC,+B;MAAA,mC;MACI,aAC8B,qB;K;;IAFIC,2C;MAAA,0C;QAAA,yB;;MAAA,mC;K;;Igm9  
GkC,oD;MAA2C,uB;MAAjB,gB;MAC5D,sBAAGc,IIKmCU,I;M0KIC1C,iBAAmC,YAAO,CAAX,GAAc,SAAS,  
IAAvB,GAAiC,SAAS,I;MACzE,cAA4B,cAA5B,GAAqC,KIKiCK,I0KjC1C,GAAqD,mB;K;gDAErD,Y;MAAkC,  
qB;K;iDAEIC,Y;MACI,YAAy,W;MACZ,IAAI,UAAAS,mBAAb,C;QACI,IAAI,CAAC,cAAL,C;UAAc,MAAa,6B;  
QAC3B,iBAAU,K;;QAGV,4BAAQ,SAAR,I;;MAEJ,OAAa,OAAAN,KAAM,C;K;;IAQgB,mD;MAAyC,sB;MAAjB,

gB;MACzD,sBAAGC,I;MACHC,iBAAMC,YAAO,CAAX,GAAc,SAAS,IAAvB,GAAiC,SAAS,I;MACzE,cAA4B,c  
AAJ,GAAa,KAAb,GAAwB,mB;K;+CAEhD,Y;MAAkC,qB;K;+CAEIC,Y;MACI,YAAY,W;MACZ,IAAI,UAAS,  
mBAAb,C;QACI,IAAI,CAAC,cAAL,C;UAAc,MAAa,6B;QAC3B,iBAAU,K;QAGV,4BAAQ,SAAR,I;MAEJ,OA  
AO,K;K;;IAQuB,oD;MAA4C,uB;MAAlB,gB;MAC5D,sBAAiC,I;MACjC,iBAAMC,uBAAO,CAAX,GAAc,sBAA  
S,IAAT,MAAd,GAAiC,sBAAS,IAAT,M;MACHC,cAA6B,cAAJ,GAAa,KAAb,GAAwB,mB;K;gDAEjD,Y;MAAk  
C,qB;K;iDAEIC,Y;MACI,YAAY,W;MACZ,IAAI,cAAS,mBAAT,CAAJ,C;QACI,IAAI,CAAC,cAAL,C;UAAc,M  
AAa,6B;QAC3B,iBAAU,K;QAGV,8BAAQ,SAAR,C;;MAEJ,OAAO,K;K;;IC9DX,oD;MA6CA,uC;MatCI,IAAI,  
SAAQ,CAAZ,C;QAAe,MAAa,gCAAyB,wBAAzB,C;MAC5B,IAAI,SAAQ,WAAZ,C;QAA2B,MAAa,gCAAyB,w  
EAAzB,C;MAG5C,aAGyB,K;MAEzB,YAGuF,OAA/D,0BAA0B,K3KeR,I2KfIB,EAAc,Y3KepB,I2KfIB,EAAy  
D,IAAZD,CAA+D,C;MAEvF,YAGuB,I;K;yCAEvB,Y;MAAwC,mCAAwB,UAAxB,EAA+B,SAA/B,EAAqC,SAA  
rC,C;K;wCAExC,Y;MAMqC,OAAI,YAAO,CAAX,GAAc,AAAQ,SAAtB,GAAgC,AAAQ,S;K;uCAE7E,iB;MACI,i  
DAA6B,kBAAa,KAAM,UAAAnB,KAC7B,eAAS,KAAM,MAAf,IAAwB,cAAQ,KAAM,KAAtC,IAA8C,cAAQ,KA  
AM,KAD/B,CAA7B,C;K;yCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,OAAM,OAAK,U3KPG,I2KOR,U  
AAkB,S3KPV,I2KOR,KAAN,SAAqC,SAArC,I;K;yCAE5B,Y;MAAkC,OAAI,YAAO,CAAX,GAAc,oBAAE,UA  
AF,+BAAU,SAAV,eAAqB,SAAnC,GAA8C,oBAAE,UAAf,qCAAgB,SAAhB,gBAA4B,CAAC,SAAD,IAA5B,C;  
K;IAEHF,qC;MAAA,yC;K;kEACI,sC;MAQ2F,2BAAgB,UAAhB,EAA4B,QAA5B,EAAc,IAAtC,C;K;;IAT/F,iD;  
MAAA,gD;QAAA,+B;;MAAA,yC;K;;IAiBA,mD;MA6CA,sC;MatCI,IAAI,SAAQ,CAAZ,C;QAAe,MAAa,gCAA  
yB,wBAAzB,C;MAC5B,IAAI,SAAQ,WAAZ,C;QAA2B,MAAa,gCAAyB,wEAAzB,C;MAG5C,aAGwB,K;MAEx  
B,YAGuB,0BAA0B,KAA1B,EAAiC,YAAjC,EAA+C,IAA/C,C;MAEvB,YAGuB,I;K;wCAEvB,Y;MAAuC,kCAA  
uB,UAAvB,EAA8B,SAA9B,EAAoC,SAAPC,C;K;uCAEvC,Y;MAMqC,OAAI,YAAO,CAAX,GAAc,AAAQ,SAAt  
B,GAAgC,AAAQ,S;K;sCAE7E,iB;MACI,gDAA4B,kBAAa,KAAM,UAAAnB,KAC5B,eAAS,KAAM,MAAf,IAAw  
B,cAAQ,KAAM,KAAtC,IAA8C,cAAQ,KAAM,KADhC,CAA5B,C;K;wCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf  
,GAAwB,OAAM,MAAK,UAAAL,QAAa,SAAb,IAAN,SAA2B,SAA3B,I;K;wCAE5B,Y;MAAkC,OAAI,YAAO,CA  
AX,GAAgB,UAAf,qBAAU,SAAV,cAAqB,SAAnC,GAAgD,UAAf,2BAAgB,SAAhB,eAA4B,CAAC,SAAD,IAA  
5B,C;K;IAEHF,oC;MAAA,wC;K;iEACI,sC;MAQwF,0BAAe,UAAf,EAA2B,QAA3B,EAAqC,IAArC,C;K;;IAT5F  
,gD;MAAA,+C;QAAA,8B;;MAAA,wC;K;;IAiBA,oD;MA6CA,uC;MatCI,IAAI,gBAAJ,C;QAAgB,MAAa,gCAAy  
B,wBAAzB,C;MAC7B,IAAI,sCAAJ,C;QAA4B,MAAa,gCAAyB,yEAAzB,C;MAG7C,aAGyB,K;MAEzB,YAGw  
B,4BAA0B,KAA1B,EAAiC,YAAjC,EAA+C,IAA/C,C;MAExB,YAGwB,I;K;yCAExB,Y;MAAwC,mCAAwB,UA  
AxB,EAA+B,SAA/B,EAAqC,SAArC,C;K;wCAExC,Y;MAMqC,OAAI,uBAAO,CAAX,GAAc,2BAAQ,SAAR,K  
AAAd,GAAgC,2BAAQ,SAAR,K;K;uCAErE,iB;MACI,iDAA6B,kBAAa,KAAM,UAAAnB,KAC7B,mBAAS,KAAM,  
MAAf,KAAwB,kBAAQ,KAAM,KAAAd,CAAxB,IAA8C,kBAAQ,KAAM,KAAAd,CADjB,CAA7B,C;K;yCAGJ,Y;  
MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,iCAAM,iCAAM,eAAW,8BAAW,EAAx,CAAX,CAAN,MAAoC,cAA  
U,6BAAU,EAAV,CAAV,CAApC,CAAN,MAAuE,cAAU,6BAAU,EAAV,CAAV,CAAvE,CAAiG,Q;K;yCAE7H,  
Y;MAAkC,OAAI,uBAAO,CAAX,GAAgB,UAAf,qBAAU,SAAV,yBAAqB,SAArB,WAAAd,GAAgD,UAAf,2BA  
AgB,SAAhB,yBAA6B,SAAD,AA5B,W;K;IAEHF,qC;MAAA,yC;K;kEACI,sC;MAQ4F,2BAAgB,UAAhB,EAA4  
B,QAA5B,EAAc,IAAtC,C;K;;IAThG,iD;MAAA,gD;QAAA,+B;;MAAA,yC;K;;6CCIKa,iB;MAKkD,+BAAS,  
UAAT,UAAkB,wBAAS,iBAAT,M;K;oCAEpE,Y;MAKgC,oCAAQ,iBAAR,K;K;;8CAuBhC,iB;MAKkD,+BAAS  
,UAAT,UAAkB,wBAAQ,iBAAR,K;K;qCAEpE,Y;MAKgC,oCAAS,iBAAT,M;K;;ICxDiB,8C;MACjD,4B;MACA,  
0C;K;oEADA,Y;MAAA,2B;K;2EACA,Y;MAAA,kC;K;uCAGA,iB;MACI,OAAO,0CAAqC,kBAAa,KAAM,UAA  
nB,KAC/B,mBAAS,KAAM,MAAf,KAAwB,0BAAgB,KAAM,aAAtB,CADo,CAAhC,C;K;yCAIX,Y;MACI,OAA  
W,cAAJ,GAAe,EAAf,GAAuB,MAAW,SAAN,UAAAM,CAAX,QAAqC,SAAb,iBAAa,CAArC,I;K;yCAGiC,Y;MA  
AkC,OAAE,UAAf,qBAAU,iB;K;;IAGhD,kC;MAM6E,2BAAgB,SAAhB,EAAc,IAAtB,C;K;IAMjB,qD;MACxD  
,4B;MACA,0C;K;2EADA,Y;MAAA,2B;K;kFACA,Y;MAAA,kC;K;8CAGA,iB;MACI,OAAO,iDAAuC,kBAAa,K  
AAM,UAAAnB,KACtC,mBAAS,KAAM,MAAf,KAAwB,0BAAgB,KAAM,aAAtB,CADc,CAAvC,C;K;gDAIX,Y;  
MACI,OAAW,cAAJ,GAAe,EAAf,GAAuB,MAAW,SAAN,UAAAM,CAAX,QAAqC,SAAb,iBAAa,CAArC,I;K;gD  
AGiC,Y;MAAkC,OAAE,UAAf,sBAAW,iB;K;;IAGjD,wC;MAQiF,kCAAuB,SAAvB,EAA6B,IAA7B,C;K;;0DA  
Y7E,iB;MAA2C,qCAAiB,UAAjB,EAAwB,KAAxB,KAAkC,8BAAiB,KAAjB,EAAwB,iBAAxB,C;K;iDAC7E,Y;  
MAAkC,QAAC,8BAAiB,UAAjB,EAAwB,iBAAxB,C;K;;IAcR,gD;MAI3B,gBAAqB,K;MACrB,uBAA4B,Y;K;0F

ACD,Y;MAAQ,oB;K;iGACD,Y;MAAQ,2B;K;2DAE1C,gB;MAA+D,YAAK,C;K;mDAEpE,iB;MAAgD,gBAAS,a  
AAT,IAAmB,SAAS,oB;K;0CAC5E,Y;MAAkC,SAAE,iBAAU,oBAAZ,C;K;yCAEIC,iB;MACI,OAAO,4CAA+B,  
kBAaA,KAAM,UAAAnB,KAC9B,kBAAU,KAAM,SAAhB,IAA0B,yBAAiB,KAAM,gBADnB,CAA/B,C;K;2CAIX  
,Y;MACI,OAAW,cAAJ,GAAe,EAaf,GAAuB,MAAY,SAAP,aAAO,CAAZ,QAAuC,SAAd,oBAAc,CAAvC,I;K;2  
CAGIC,Y;MAAkC,OAAE,aAAf,qBAAW,oB;K;;IAGjD,oC;MAOqF,6BAAkB,SAaIB,EAAwB,IAAxB,C;K;IAQr  
D,iD;MAI5B,gBAAqB,K;MACrB,uBAA4B,Y;K;2FACD,Y;MAAQ,oB;K;K;GACD,Y;MAAQ,2B;K;sDAE1C,gB;  
MAA8D,YAAK,C;K;oDAEnE,iB;MAAgD,gBAAS,aAAT,IAAmB,QAAQ,oB;K;2CAC3E,Y;MAAkC,SAAE,gBA  
AS,oBAAX,C;K;0CAEIC,iB;MACI,OAAO,6CAAgC,kBAaA,KAAM,UAAAnB,KAC/B,kBAAU,KAAM,SAAhB,I  
AA0B,yBAAiB,KAAM,gBADiB,CAAhC,C;K;4CAIX,Y;MACI,OAAW,cAAJ,GAAe,EAaf,GAAuB,MAAY,SAAP  
P,aAAO,CAAZ,QAAuC,SAAd,oBAAc,CAAvC,I;K;4CAGIC,Y;MAAkC,OAAE,aAAf,sBAAy,oB;K;;IAGID,wC;  
MAO4E,8BAAmB,SAAnB,EAAYB,IAAZB,C;K;IAQ9C,+C;MAI1B,gBAAqB,K;MACrB,uBAA4B,Y;K;yFACF,Y  
;MAAQ,oB;K;gGACD,Y;MAAQ,2B;K;0DAEzC,gB;MAA6D,YAAK,C;K;kDAEIE,iB;MAA+C,gBAAS,aAAT,IA  
AmB,SAAS,oB;K;yCAC3E,Y;MAAkC,SAAE,iBAAU,oBAAZ,C;K;wCAEIC,iB;MACI,OAAO,2CAA8B,kBAaA,  
KAAM,UAAAnB,KAC7B,kBAAU,KAAM,SAAhB,IAA0B,yBAAiB,KAAM,gBADpB,CAA9B,C;K;0CAIX,Y;MA  
CI,OAAW,cAAJ,GAAe,EAaf,GAAuB,MAAY,SAAP,aAAO,CAAZ,QAAuC,SAAd,oBAAc,CAAvC,I;K;0CAGIC,  
Y;MAAkC,OAAE,aAAf,qBAAW,oB;K;;IAGjD,oC;MAOkF,4BAAiB,SAAjB,EAAuB,IAAvB,C;K;IASnD,gD;M  
AI3B,gBAAqB,K;MACrB,uBAA4B,Y;K;0FACF,Y;MAAQ,oB;K;iGACD,Y;MAAQ,2B;K;qDAEzC,gB;MAA4D,  
YAAK,C;K;mDAEjE,iB;MAA+C,gBAAS,aAAT,IAAmB,QAAQ,oB;K;0CAC1E,Y;MAAkC,SAAE,gBAAS,oBA  
AX,C;K;yCAEIC,iB;MACI,OAAO,4CAA+B,kBAaA,KAAM,UAAAnB,KAC9B,kBAAU,KAAM,SAAhB,IAA0B,y  
BAAiB,KAAM,gBADnB,CAA/B,C;K;2CAIX,Y;MACI,OAAW,cAAJ,GAAe,EAaf,GAAuB,MAAY,SAAP,aAAO  
,CAAZ,QAAuC,SAAd,oBAAc,CAAvC,I;K;2CAGIC,Y;MAAkC,OAAE,aAAf,sBAAy,oB;K;;IAGID,wC;MAOyE  
,6BAAkB,SAaIB,EAAwB,IAAxB,C;K;oFAGzE,8B;MAQI,0BAAmB,2BAAS,OAAT,C;K;oFAEvB,8B;MASI,0B  
AAmB,2BAAS,OAAT,C;K;IAEvB,+C;MACI,IAAI,CAAC,UAAL,C;QAAiB,MAAM,gCAAYB,iCAA8B,IAA9B,i  
BAAzB,C;K;ICxQ3B,gC;MAcW,Q;MADP,IAAI,CAAC,6BAAW,KAAX,CAAL,C;QAAwB,MAAM,uBAAmB,sC  
pFjBzC,oBoFiByC,CAAnB,C;;MAC9B,OAAO,sD;K;IAMX,oC;MAAkC,Q;MAA9B,OAAW,6BAAW,KAAX,CA  
AJ,GAAuB,sDAAvB,GAAuC,I;K;;;ICvBhB,yC;MA2B9B,uC;MA1BA,wB;MAIA,gB;M9LQA,IAAI,E8LDS,iB  
AAY,IAAb,MAAuB,iBAAvB,C9LCR,CAAJ,C;QACI,c8LDQ,iBAAy,IAAhB,GACI,8CADJ,GAGI,sCAA0B,aAA  
1B,qC;Q9LDR,MAAM,gCAAYB,OAAQ,WAAJ,C;;K;yC8LKV,Y;MAAwC,Q;MAAA,oB;MACpC,iB;QAD8B,O  
ACtB,G;WACR,oD;QAF8B,OAEF,SAAL,SAAK,C;WAC5B,6C;QAH8B,OAGd,iBAAK,SAAL,C;WACHB,8C;Q  
AJ8B,OAIb,kBAAM,SAAN,C;;QAJa,mC;K;IAOIC,qC;MAAA,yC;MACI,YAGqC,oBAAgB,IAAhB,EAASB,IAAt  
B,C;K;iGAQJ,Y;MAAQ,gB;K;4DAEzC,gB;MAOI,8DAAqC,IAArC,C;K;gEAej,gB;MAMI,uDAA8B,IAA9B,C;K  
;4DAEj,gB;MAMI,wDAA+B,IAA/B,C;K;;IArCR,iD;MAAA,gD;QAAA,+B;MAAA,yC;K;;2CarCJ,Y;MAWI,oB  
;K;2CAXJ,Y;MAeI,gB;K;6CAfJ,0B;MAAA,2BAWI,8CAXJ,EAeI,kCAfJ,C;K;yCAAA,Y;MAAA,c;MAWI,yD;M  
AIA,qD;MAfJ,a;K;uCAAA,iB;MAAA,4IAWI,4CAXJ,IAeI,oCAfJ,I;K;ICLA,kC;MAAA,e;MAAA,iB;MAAA,uB;  
K;IAAA,gC;MAAA,mC;O;MAYI,4D;MAKA,8C;MAKA,gD;K;;IAVA,2C;MAAA,sB;MAAA,mC;K;;IAKA,oC;M  
AAA,sB;MAAA,4B;K;;IAKA,qC;MAAA,sB;MAAA,6B;K;;IAtBJ,4B;MAAA,mG;K;;IAAA,iC;MAAA,a;AAA,  
W;UAAA,wC;aAAA,I;UAAA,iC;aAAA,K;UAAA,kC;;UAAA,6D;;K;;6ECAA,yB;MAAA,4F;MAAA,2B;QASI,M  
AAM,mCAA8B,0EAA9B,C;O;KATV,C;ICkCA,+D;MAAw,Q;MAAP,OAAO,8CAA0,KAAP,EAAC,UAAAd,EAA0  
B,QAA1B,oC;K;IAGX,kC;MAIiB,Q;MAAb,wBAaA,KAAb,gB;QAAa,WAAA,KAAb,M;QACI,yBAAO,IAAP,C;;  
MACJ,OAAO,S;K;mFAGX,qB;MAGwD,gCAAO,EAAP,C;K;qFAExD,4B;MAG4E,OAAA,yBAAO,KAAP,CALp  
B,gBAAO,EAAP,C;K;qFAOxD,4B;MAGmE,OAAA,yBAAO,KAAP,CAVX,gBAAO,EAAP,C;K;IAaxD,wD;MAE  
Q,sB;QAAqB,yBAAO,UAAU,OAAP,CAAP,C;WACrB,sD;QAA4B,yBAAO,OAAP,C;WAC5B,2B;QAAmB,yBA  
AO,kBAAP,C;;QACX,yBAAE,SAAR,OAAQ,CAAF,C;K;InL7EhB,+B;MAY6B,kBAAiB,QAAQ,SAAR,EAAC,EA  
Ad,C;MACH,IX0EE,WW1EE,GAAK,CAAT,C;QAAY,MAAM,gCAAYB,oEAAzB,C;MADtB,OX4EO,W;K;IWvE  
X,wC;MAGBW,Q;MAAA,qCAAiB,KAajB,C;MAAA,iB;QAA2B,MAAM,gCAAYB,8BAAO,SAAP,4CAA+C,KA  
AxE,C;;MAAxC,OAAO,I;K;IAGX,qC;MAY6B,kBAAiB,QAAQ,SAAR,EAAC,EAAd,C;MAAP,OXmEqB,WWnE  
a,IAAM,CXmEjC,GAAqB,WAArB,GAA+B,I;K;IWhE1C,8C;MAGBI,WAAW,KAAX,C;MAC4B,kBAArB,QAAQ  
,SAAR,EAAC,KAAd,C;MAAP,OX+CqB,WW/CgB,IAAM,CX+CpC,GAAqB,WAArB,GAA+B,I;K;IW5C1C,gC;



MAWI,IAAY,CAAR,8BAAW,CAAf,C;QACI,OAAO,YAAM,SAAN,C;;MAEX,MAAM,gCAAYB,SAAM,SAAN,4BAAzB,C;K;IAGV,yC;MAkBW,Q;MANP,IAAI,EAAU,CAAV,sBAaAa,EAAb,CAAJ,C;QACI,MAAM,gCAAYB,oBAaIB,KAAjB,4CAAzB,C;;MAEV,IAAI,YAAO,CAAP,IAAY,aAAQ,KAAxB,C;QACI,MAAM,gCAAYB,WAAQ,SAAR,mDAAwD,KAAjF,C;;MAEH,IAAI,YAAO,EAAX,C;QACH,mBAAM,SAAN,C;;QAEA,0BAAM,SAAN,IAAa,EAAb,C;;MAHJ,W;K;IAuFJ,8B;MAWSc,+B;K;0EAEtC,4B;MAM8D,OAAK,oBAAL,SAAK,CAAL,GAAB,K;K;IAEHf,gD;MAQoC,0B;QAAA,aAAsB,K;MACtD,IAAI,cAAQ,KAAZ,C;QAAMB,OAAO,I;MAC1B,IAAI,CAAC,UAAAL,C;QAAiB,OAAO,K;MAExB,gBAaQb,cAAL,SAAK,C;MACrB,iBAaUaB,cAAN,KAAM,C;MAEHb,yBAaAa,U;MAAb,U;QAA2B,OfRMyB,oBEqMzB,SFrMyB,CAAY,cAfrB,YAAY,CAAZ,CEoNhB,KFrMyB,oBEqMI,UFrMJ,CAAY,cAfrB,YAAY,CAAZ,C;;MEoNID,W;K;IAGJ,gC;MAGyC,QAAQ,cAAA,sCAAK,cAAL,EAAoB,sCAAK,cAAzB,CAAR,6B;K;IoL3OzC,6C;MAe6B,4B;QAAA,eAAuB,G;MACHD,wCAAsB,EAAtB,EAA0B,YAA1B,C;K;IAEJ,mE;MAKwC,yB;QAAA,YAAoB,E;MAAI,4B;QAAA,eAAuB,G;MIMEnF,IAAI,CmBwR+C,CAAC,Q+KzR5C,Y/KyR4C,CnBxRpD,C;QACI,ckMFiC,wC;QIMGjC,MAAM,gCAAYB,OAAQ,WAAjC,C;;MkMFV,cAAY,gB;MAEC,yBAAS,mBAAS,YAAA,SAAU,OAAV,EAAMB,OAAM,KAAzB,CAAT,I;MAAT,wBAaID,kBAaKB,SAAlB,C;MA2E9D,gBAAGb,iBA3ET,OA2ES,C;M5Lg7CT,kBAaOb,gB;MAoSd,gB;MADb,YAAY,C;MACC,O4L/xDN,O5L+xDM,W;kBAAb,OAAa,cAAb,C;QAAA,sB;QA1RsB,U;QAAA,cA0RT,oBAAMB,cAAnB,EAAMb,sBAAnB,U;Q4L/sDIB,kB;;YAHA,CAAC,YAAS,CAAT,IAAc,qBAaf,KAA4C,Q5LktDG,I4LItDH,C;UAC5C,a;;UAEA,4B;UA/E+B,uB;;YhLgHzB,kC;YAAA,wBZ8qDyC,IY9qDzC,C;YAAA,qB;YAAA,oB;YAAA,oB;YAd,gE;cACI,IgLjHkD,CAAI,aAAH,UhLiHrC,YZ6qDqC,IY7qDrC,YAAK,OAAL,EgLjHqC,CAAG,ChLiHtD,C;gBACI,sBAAO,O;gBAAP,wB;;YAGR,sBAAO,E;;UgLrHH,iD;UAGI,gCAA2B,EAA3B,C;YAHJ,2BAGqC,I;iBACjC,IAAK,a5L0xD0C,I4L1xD1C,gBAAYb,uBAAZB,CAAL,C;YAJJ,2B5L8xDmD,IO1kDsB,WqLhNI,0BAAuC,mBAAvC,IrLgNJ,C;;YqLpNzE,2BAKY,I;;UA0ER,iE/LND,yB+LMC,4B5L+sD+C,I;;QA1RpB,8B;UAA6C,6B;;M4LrgDhF,OakFK,S5Lo7CE,W4Lp7CF,EAAO,mBAAc,kBAAd,CAAP,EAA0C,IAA1C,CACA,W;K;IAxET,+B;MAGByC,gCAAc,EAAd,C;K;IAEzC,6C;MAGgC,yB;QAAA,YAAoB,E;MAM3C,Q;MALL,cAAY,gB;M5LurBL,kBAAS,gB;MA2FA,U;MAAA,S4LhxBM,O5LgxBN,W;MAAhB,OAAgB,gBAAhB,C;QAAGB,2B;QAAM,Ia7hB6B,C AAC,Qb6hBhB,Oa7hBgB,Cb6hB9B,C;UAAwB,WAAy,WAAI,OAAJ,C;;M4L9wBrD,kB5L+wBE,W;MAMrBA,oBAAM,iBAaAa,qCAAwB,EAAXB,CAAb,C;MAuEA,U;MAAA,+B;MAAb,OAAa,gBAAb,C;QAAA,wB;QACT,aAAY,uBAAC,IAAd,E;;M4L5gDhB,sBAAsB,CAGjB,oB5L0gDE,a4L1gDF,CAHiB,mBAGf,C;MAEP,yBAAS,mBAAS,YAAA,SAAU,OAAV,EAAMB,OAAM,KAAzB,CAAT,I;MAAT,wBAaID,kBAaKB,SAAlB,C;MAMc9D,gBAAGb,iBAAnCT,OAmCS,C;M5Lg7CT,oBAaOb,gB;MAoSd,kB;MADb,YAAY,C;MACC,S4LvvdDN,O5LuvDM,W;MAAb,OAAa,gBAAb,C;QAAA,0B;QA1RsB,U;QAAA,cA0RT,oBAAMB,cAAnB,EAAMb,sBAAnB,U;Q4L/sDIB,kB;Q5Lq7C2B,c4Lx7C3B,CAAC,YAAS,CAAT,IAAc,qBAaf,KAA4C,Q5LktDG,M4LItDH,C5Lw7CjB,G4Lv7C3B,I5Lu7C2B,G4Lr7C3B,oBAxCmG,Q5LuvDpD,M4LvDoD,kBAwCnG,Y/LND,yB+LMC,4B5L+sD+C,MA1RpB,U;UAA6C,+B;;M4L79ChF,OA0CK,S5Lo7CE,a4Lp7CF,EAAO,mBAAc,kBAAd,CAAP,EAA0C,IAA1C,CACA,W;K;IAjCI,8C;MAAA,qB;QAEG,IAAG,QAAG,EAAG,CAAG,C;UAEQ,IAAA,EAAG,OAAG,GAAY,cAAO,OAAnB,C;YAHZ,OAGyC,c;;YAHZC,OAIoB,E;;UAJpB,OAoy,iBAAS,E;O;K;IAfjC,0C;MAKgC,sB;QAAA,SAaIB,M;MAC7C,OAYK,eAXA,OADL,uBACK,EAAl,4BAAJ,CAWA,EAAa,IAAb,C;K;IAET,gC;MAAwC,uB;;QhLkDtB,gC;QAAA,gC;QAAA,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IgLnD+C,CAAI,aAAH,UhLmDIC,iCAAK,KAAAL,EgLnDkC,CAAG,ChLmDnD,C;YACI,sBAAO,K;YAAP,wB;;QAGR,sBAAO,E;;Mf5CA,4B;M+LX6B,OA8C,OAAM,EAAV,GAAC,gBAAd,GAA0B,E;K;IAGpF,wC;MAAkB,W;K;IAC9B,oD;MAAA,uB;QAaKB,wBAAS,I;O;K;IAFvC,mC;MACI,IAAA,M/KgMgD,YAAU,C+KhM1D,C;QAD4C,OACxB,wB;;QADwB,OAEPc,kC;K;mBAGZ,yB;M5L86CA,+D;MAoSA,wE;M4LItDA,sF;QAKI,gBAAGb,2B;Q5Lg7CT,kBAaOb,gB;QAoSd,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAA,sB;UA1RsB,U;UAAA,cA0RT,oBAAMB,cAAnB,EAAMb,sBAAnB,U;U4L/sDIB,kB;U5Lq7C2B,c4Lx7C3B,CAAC,YAAS,CAAT,IAAc,qBAaf,KAA4C,Q5LktDG,I4LItDH,C5Lw7CjB,G4Lv7C3B,I5Lu7C2B,G4Lr7C3B,sC5L+sD+C,I4L/sD/C,a/LND,yB+LMC,4B5L+sD+C,IA1RpB,U;YAA6C,6B;;Q4Lz7ChF,OAMK,S5Lo7CE,W4Lp7CF,EAAO,mBAAc,kBAAd,CAAP,EAA0C,IAA1C,CACA,W;O;KAbT,C;6E5EgSA,0B;MAGmE,OAAA,SAAK,gBAAO,GAAP,C;K;qFAExE,yB;MAAA,yD;MAAA,gC;QAO2B,gBAAhB,oB;QAAsB,anHrU7B,W;QmHqUA,OnHpUO,SmHoUqC,W;O;KAPhD,C;uFAUA,yB;MAAA,iE;MAAA,0C;QAQmC,gBAAXB,mBAAC,QAAd,C;QAA8B,anHhVrC,W;QmHgVA,OnH/uo,SmH+U6C,W;O;KARxD,C;I

AWA,oC;MAliB,Q;MAAb,wBAAa,KAAb,gB;QAAa,WAAA,KAAb,M;QACI,yBAAO,IAAP,C;;MACJ,OAAO,S;  
K;IAGX,oC;MAliB,Q;MAAb,wBAAa,KAAb,gB;QAAa,WAAA,KAAb,M;QACI,yBAAO,IAAP,C;;MACJ,OAAO,  
S;K;qFAGX,qB;MAG8D,gCAAO,EAAP,C;K;qFAE9D,4B;MAGkF,OAAA,yBAAO,KAAP,CALpB,gBAAO,EA  
P,C;K;qFAO9D,4B;MAG4E,OAAA,yBAAO,KAAP,CAVd,gBAAO,EAAP,C;K;qFAY9D,4B;MAGyE,OAAA,yB  
AAO,KAAP,CAfX,gBAAO,EAAP,C;K;qFAiB9D,4B;MAG8E,OAAA,yBAAO,KAAP,CAPhB,gBAAO,EAAP,C  
;K;qFAsB9D,4B;MAGyE,OAAA,yBAAO,KAAP,CAzBX,gBAAO,EAAP,C;K;qFA2B9D,4B;MAG4E,OAAA,yB  
AAO,KAAP,CA9Bd,gBAAO,EAAP,C;K;I5H/a9D,iC;MAK0C,iCAAqB,EAARb,C;K;IAE1C,0C;MAQmB,Q;MAA  
A,qBAAL,SAAK,EAAY,KAAZ,C;MAAL,iB;QAA2B,OAAO,I;;MAA5C,UAAU,I;MACV,IAAI,MAAM,sCAAK,  
UAAX,IAAwB,MAAM,sCAAK,UAAvC,C;QAAkD,OAAO,I;MACzD,OAAW,OAAJ,GAAL,C;K;IAGf,kC;MAK4  
C,kCAAsB,EAAtB,C;K;IAE5C,2C;MAQmB,Q;MAAA,qBAAL,SAAK,EAAY,KAAZ,C;MAAL,iB;QAA2B,OAA  
O,I;;MAA5C,UAAU,I;MACV,IAAI,MAAM,uCAAM,UAZ,IAAyB,MAAM,uCAAM,UAAzC,C;QAAoD,OAAO  
,I;MAC3D,OAAW,QAAJ,GAAL,C;K;IAGf,gC;MAKwC,gCAAOB,EAAPb,C;K;IAExC,yC;MAQI,WAAW,KAAX  
,C;MAEA,aAAa,SAAK,O;MACIB,IAAI,WAAU,CAAd,C;QAAiB,OAAO,I;MAExB,S;MACA,c;MACA,S;MAEA,  
gBAAGb,qBAAK,CAAL,C;MACHb,IAAI,YAAY,EAAbB,C;QACI,IAAI,WAAU,CAAd,C;UAAiB,OAAO,I;QAE  
xB,QAAQ,C;QAER,IAAI,cAAa,EAajB,C;UACI,aAAa,I;UACb,QAAQ,W;eACL,IAAI,cAAa,EAajB,C;UACH,aA  
Aa,K;UACb,QAAQ,W;;UAER,OAAO,I;;QAEX,QAAQ,C;QACR,aAAa,K;QACb,QAAQ,W;;MAIZ,uBAAuB,S;M  
AEvB,qBAaQb,gB;MACrB,aAAa,C;MACb,aAAU,KAAV,MAAsB,MAAtB,M;QACI,YAAY,QAAQ,qBAAK,CA  
AL,CAAR,EAaiB,KAAjB,C;QAEZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,IAAI,SAAS,cAAb,C;UACI,I  
AAI,mBAaKb,gBAAtB,C;YACI,iBAaiB,QAAQ,KAAR,I;YAEjB,IAAI,SAAS,cAAb,C;cACI,OAAO,I;;YAGX,O  
AAO,I;;QAIIf,6BAAU,KAAV,C;QAEA,IAAI,UAAS,QAAQ,KAAR,IAAT,CAAJ,C;UAA4B,OAAO,I;QAEnC,kB  
AAU,KAAV,I;;MAGJ,OAAW,UAAJ,GAAGb,MAAhB,GAA4B,CAAC,MAAD,I;K;IAGvC,iC;MAK0C,iCAAqB,  
EAARb,C;K;IAE1C,0C;MAQI,WAAW,KAAX,C;MAEA,aAAa,SAAK,O;MACIB,IAAI,WAAU,CAAd,C;QAAiB,  
OAAO,I;MAExB,S;MACA,c;MACA,S;MAEA,gBAAGb,qBAAK,CAAL,C;MACHb,IAAI,YAAY,EAAbB,C;QAC  
I,IAAI,WAAU,CAAd,C;UAAiB,OAAO,I;QAExB,QAAQ,C;QAER,IAAI,cAAa,EAajB,C;UACI,aAAa,I;UACb,gC  
;eACG,IAAI,cAAa,EAajB,C;UACH,aAAa,K;UACb,6B;;UAEA,OAAO,I;;QAEX,QAAQ,C;QACR,aAAa,K;QAC  
b,6B;;MAIJ,2C;MAEA,qBAaQb,gB;MACrB,e;MACA,aAAU,KAAV,MAAsB,MAAtB,M;QACI,YAAY,QAAQ,q  
BAAK,CAAL,CAAR,EAaiB,KAAjB,C;QAEZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,IAAI,uBAAS,cAA  
T,KAAJ,C;UACI,IAAI,uBAaKb,gBAaIB,CAAJ,C;YACI,iBAaiB,8BAAQ,KAAR,E;YAEjB,IAAI,uBAAS,cAAT  
,KAAJ,C;cACI,OAAO,I;;YAGX,OAAO,I;;QAIIf,6CAAU,KAAV,E;QAEA,IAAI,uBAAS,8BAAQ,KAAR,EAAT,  
KAAJ,C;UAA4B,OAAO,I;QAEnC,6CAAU,KAAV,E;;MAGJ,OAAW,UAAJ,GAAGb,MAAhB,GAA6B,MAAD,a;  
K;IAIvC,kC;MAAyD,MAAM,0BAAsB,6BAA0B,KAA1B,MAAtB,C;K;uEyBhI/D,yB;MAAA,oC;MAAA,uC;QAI  
I,iBAaiB,C;QACjB,eAAe,mBAAS,CAAT,I;QACf,iBAaiB,K;QAEjB,OAAO,cAAc,QAARb,C;UACI,YAAGb,CA  
AC,UAAL,GAaiB,UAAjB,GAaiC,Q;UAC7C,YAAY,UAAU,iCAAk,KAAL,EAAV,C;UAEZ,IAAI,CAAC,UAA  
L,C;YACI,IAAI,CAAC,KAAL,C;cACI,aAAa,I;;cAEb,0BAAc,CAAd,I;;YAEJ,IAAI,CAAC,KAAL,C;cACI,K;;cA  
EA,sBAAY,CAAZ,I;;QAIZ,OAAO,8BAAY,UAAZ,EAawB,WAAW,CAAX,IAAxB,C;O;KAZBX,C;yEA4BA,yB  
;MAAA,8B;MA5BA,oC;MA4BA,uC;QAIK,Q;QAAsB,kBAAtB,2D;QA5BD,iBAaiB,C;QACjB,eAAe,qBAAS,CA  
AT,I;QACf,iBAaiB,K;QAEjB,OAAO,cAAc,QAARb,C;UACI,YAAGb,CAAC,UAAL,GAaiB,UAAjB,GAaiC,Q;  
UAC7C,YAsBwB,SAtBZ,CAAU,mCAAk,KAAL,EAAV,C;UAEZ,IAAI,CAAC,UAAL,C;YACI,IAAI,CAAC,KA  
AL,C;cACI,aAAa,I;;cAEb,0BAAc,CAAd,I;;YAEJ,IAAI,CAAC,KAAL,C;cACI,K;;cAEA,sBAAY,CAAZ,I;;QAW  
Z,OAPo,gCAAY,UAAZ,EAawB,WAAW,CAAX,IAAxB,CAOGC,W;O;KAJ3C,C;iFAMA,yB;MAAA,mD;MAA  
A,oC;MAAA,uC;QAIuB,UAAL,MAAK,EAAL,MAAK,EAAL,M;QAAK,mBAAL,SAAK,C;QAAL,mB;QAAA,k  
B;QAAA,kB;QAAAd,0D;UACI,IAAI,CAAC,UAAU,iCAAk,KAAL,EAAV,CAAL,C;YACI,OAAO,8BAAY,KAAZ  
,EAAMb,gBAAnB,C;QAEf,OAAO,E;O;KARX,C;mFAWA,yB;MAAA,8B;MAXA,mD;MAAA,oC;MAWA,uC;Q  
AIK,Q;QAAsB,kBAAtB,2D;QAAsB,oB;;UAXJ,kC;UAAA,qBAAL,WAAK,C;UAAL,qB;UAAA,oB;UAAA,oB;U  
AAAd,0D;YACI,IAAI,CAUyB,SAVxB,CAAU,mCAAk,KAAL,EAAV,CAAL,C;cACI,mBAAO,gCAAY,KAAZ,E  
AAmB,kBAAnB,C;cAAP,qB;;UAER,mBAAO,E;;QAOP,OAA4C,2B;O;KAJhD,C;6EAMA,yB;MAAA,mD;MAA  
A,+C;MAAA,oC;MAAA,uC;QAIkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAAd,OAAc,cAAd,  
C;UAAc,uB;UACV,IAAI,CAAC,UAAU,iCAAk,KAAL,EAAV,CAAL,C;YACI,OAAO,8BAAY,CAAZ,EAae,QA

AQ,CAAR,IAAf,C;;QAEf,OAAO,E;O;KARX,C;+EAWA,yB;MAAA,8B;MAXA,mD;MAAA,+C;MAAA,oC;MAWA,uC;QAIK,Q;QAAsB,kBAAtB,2D;QAAsB,kB;;UAXT,U;UAAA,SAAa,SAAR,YAAL,WAAK,CAAQ,CAAb,W;UAAAd,OAAC,gBAAd,C;YAAc,yB;YACV,IAAI,CAUuB,SAVtB,CAAU,mCAAK,KAAL,EAAV,CAAL,C;cACI,iBAAO,gCAAY,CAAZ,EAAe,QAAQ,CAAR,IAAf,C;cAAP,mB;;UAER,iBAAO,E;;QAOP,OAAOC,yB;O;KAJ9C,C;IAMA,kC;MAhEI,iBAAiB,C;MACjB,eAAe,mBAAS,CAAT,I;MACf,iBAAiB,K;MAEjB,OAAO,cAAc,QAArB,C;QACI,YAAgB,CAAC,UAAAL,GAAiB,UAAjB,GAAiC,Q;QAC7C,YA6DgE,4BA7D1C,iCAAK,KAAL,EA6D0C,E;QA3DhE,IAAI,CAAC,UAAAL,C;UACI,IAAI,CAAC,KAAL,C;YACI,aAAa,I;;YAEb,0BAAc,CAAd,I;;UA EJ,IAAI,CAAC,KAAL,C;YACI,K;;YAEA,sBAAY,CAAZ,I;;MAkDiD,OA9CtD,8BAAY,UAAZ,EAAwB,WAAW,C AAX,IAAxB,C;K;IAGDX,kC;MAzCK,Q;MAAsB,kBAAtB,2D;MA5BD,iBAAiB,C;MACjB,eAAe,qBAAS,CAAT, I;MACf,iBAAiB,K;MAEjB,OAAO,cAAc,QAArB,C;QACI,YAAgB,CAAC,UAAAL,GAAiB,UAAjB,GAAiC,Q;QA C7C,YAkEoD,4BAIE9B,mCAAK,KAAL,EAkE8B,E;QAhEpD,IAAI,CAAC,UAAAL,C;UACI,IAAI,CAAC,KAAL, C;YACI,aAAa,I;;YAEb,0BAAc,CAAd,I;;UA EJ,IAAI,CAAC,KAAL,C;YACI,K;;YAEA,sBAAY,CAAZ,I;;MAuD qC,OAnD1C,gCAAY,UAAZ,EAAwB,WAAW,CAAX,IAAxB,CAOGC,W;K;IA8C3C,uC;MAGsE,oB;;QA3C/C,gC ;QAAA,gC;QAAL,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,CA0CsE,4BA1C3D,iCAAK,KAAL,EA0C2D ,EA1C1E,C;YACI,mBAAO,8BAAY,KAAZ,EAAMB,gBAAnB,C;YAAP,qB;;QAER,mBAAO,E;;MAuC2D,uB;K ;IAEtE,uC;MAICK,Q;MAAsB,kBAAtB,2D;MAAsB,oB;;QAXJ,kC;QAAA,wBAAL,WAAK,C;QAAL,qB;QAAA, oB;QAAA,oB;QAAd,0D;UACI,IAAI,CA+C0D,4BA/C/C,mCAAK,KAAL,EA+C+C,EA/C9D,C;YACI,mBAAO,g CAAY,KAAZ,EAAMB,kBAAnB,C;YAAP,qB;;QAER,mBAAO,E;;MA4C+C,OArCV,2B;K;IAuChD,qC;MAGoE ,kB;;QApCID,Q;QAAA,OAAa,WAAR,yBAAQ,CAAb,W;QAAd,OAAC,cAAAd,C;UAAc,uB;UACV,IAAI,CAMCk E,4BAnCvD,iCAAK,KAAL,EAAMCuD,EAnCtE,C;YACI,iBAAO,8BAAY,CAAZ,EAAe,QAAQ,CAAR,IAAf,C;Y AAP,mB;;QAER,iBAAO,E;;MAGCyD,qB;K;IAEpE,qC;MA3BK,Q;MAAsB,kBAAtB,2D;MAAsB,kB;;QAXT,U; QAAA,SAAa,WAAR,eAAL,WAAK,CAAQ,CAAb,W;QAAd,OAAC,gBAAd,C;UAAc,yB;UACV,IAAI,CAwCsD, 4BAxC3C,mCAAK,KAAL,EAwC2C,EAxC1D,C;YACI,iBAAO,gCAAY,CAAZ,EAAe,QAAQ,CAAR,IAAf,C;YA AP,mB;;QAER,iBAAO,E;;MAqC6C,OA9BV,yB;K;IAGC9C,2B;MA9FI,iBAAiB,C;MACjB,eAAe,mBAAS,CAA T,I;MACf,iBAAiB,K;MAEjB,OAAO,cAAc,QAArB,C;QACI,YAAgB,CAAC,UAAAL,GAAiB,UAAjB,GAAiC,Q;Q AC7C,mCAAsB,iCAAK,KAAL,EAAtB,E;QAEA,IAAI,CAAC,UAAAL,C;UACI,IAAI,CAAC,KAAL,C;YACI,aAA a,I;;YAEb,0BAAc,CAAd,I;;UA EJ,IAAI,CAAC,KAAL,C;YACI,K;;YAEA,sBAAY,CAAZ,I;;MAGf+B,OA5EpC,8 BAAY,UAAZ,EAAwB,WAAW,CAAX,IAAxB,C;K;yEA8EX,yB;MAAA,8B;MAAA,qC;MAAA,4B;QAI2C,Q;Q AAD,OAAuB,KAAtB,2DAAsB,CAAO,W;O;KAJxE,C;IAMA,gC;MAGoD,oB;;QAI1E7B,gC;QAAA,gC;QAAL,m B;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,wBAAW,iCAAK,KAAL,EAAX,EAJ,C;YACI,mBAAO,8BAAY ,KAAZ,EAAMB,gBAAnB,C;YAAP,qB;;QAER,mBAAO,E;;MASyC,uB;K;mFAEpD,yB;MAAA,8B;MAAA,+C ;MAAA,4B;QAIgD,Q;QAAD,OAAuB,UAAAtB,2DAAsB,CAAY,W;O;KAJIF,C;IAMA,8B;MAGkD,kB;;QApEhC, Q;QAAA,OAAa,WAAR,yBAAQ,CAAb,W;QAAd,OAAC,cAAAd,C;UAAc,uB;UACV,IAAI,wBAAW,iCAAK,KAAL,EAAX,EAJ,C;YACI,iBAAO,8BAAY,CAAZ,EAAe,QAAQ,CAAR,IAAf,C;YAAP,mB;;QAER,iBAAO,E;;M AgEuC,qB;K;+EAEID,yB;MAAA,8B;MAAA,2C;MAAA,4B;QAI8C,Q;QAAD,OAAuB,QAAtB,2DAAsB,CAAU, W;O;KAJ9E,C;IAMA,8C;MAU8C,uB;QAAA,UAAgB,E;MAO5C,Q;MANd,IAAI,SAAS,CAAb,C;QACI,MAAM, gCAAYB,oBAAiB,MAAjB,wBAAzB,C;MACV,IAAI,UAAU,SAAK,OAAAnB,C;QACI,OAAy,mBAAL,SAAK,EA AY,CAAZ,EAAe,SAAK,OAAPB,C;MAEHb,SAAS,mBAAC,MAAd,C;MACK,gBAAS,SAAK,OAAd,I;MAAd,aA AU,CAAV,iB;QACI,EAAG,gBAAO,OAAP,C;MACP,EAAG,gBAAO,SAAP,C;MACH,OAAO,E;K;IAGX,gD;M ASwC,uB;QAAA,UAAgB,E;MACnD,Q;MAAD,OAAuB,SAAtB,6DAAsB,EAAS,MAAT,EAAiB,OAAjB,CAA0B ,W;K;IAErD,4C;MAU4C,uB;QAAA,UAAgB,E;MAQ1C,Q;MAPd,IAAI,SAAS,CAAb,C;QACI,MAAM,gCAAYB, oBAAiB,MAAjB,wBAAzB,C;MACV,IAAI,UAAU,SAAK,OAAAnB,C;QACI,OAAy,mBAAL,SAAK,EAAY,CAA Z,EAAe,SAAK,OAAPB,C;MAEHb,SAAS,mBAAC,MAAd,C;MACT,EAAG,gBAAO,SAAP,C;MACW,gBAAS,S AAK,OAAd,I;MAAd,aAAU,CAAV,iB;QACI,EAAG,gBAAO,OAAP,C;MACP,OAAO,E;K;IAGX,8C;MASsC,uB; QAAA,UAAgB,E;MACjD,Q;MAAD,OAAuB,OAAtB,6DAAsB,EAAO,MAAP,EAAe,OAAf,CAAwB,W;K;2FAE nD,qB;MAWI,OAAO,qBAAGB,SAAK,OAAL,KAAe,C;K;+EAG1C,qB;MAMoD,4BAAU,C;K;sFAE9D,qB;MAM uD,0BAAS,C;K;mFAMhE,yB;MAAA,2C;MAAA,4B;QAMuD,QAAC,kB;O;KANxD,C;yFAQA,yB;MAAA,2C;M AAA,4B;QAWI,OAAO,qBAAqB,QAAL,SAAK,C;O;KAXhC,C;IAiB4D,+C;MAAA,kC;MAAS,uB;MACjE,eAAO

B,C;K;gDAEpB,Y;MAA2C,gB;MAAA,iE;MAAJ,4C;K;+CAEvC,Y;MAAyC,sBAAQ,yB;K;;IARrD,+B;MAG4D,4  
C;K;+EAQ5D,qB;MAE8C,uCAAQ,E;K;+EAeTD,mC;MASI,OA5DgD,qBAAU,CA4D1D,GAAe,cAAf,GAAMC,S;  
K;6EAEvC,yB;MAAA,2C;MAAA,0C;QASI,OAAI,kBAAJ,GAAe,cAAf,GAAMC,S;O;KATvC,C;IAeI,mC;MAAQ  
,uBAAg,mBAAS,CAAT,IAAH,C;K;IAMR,qC;MAAQ,OAAA,SAAK,OAAAL,GAAc,CAAd,I;K;IAEZ,8C;MAIuB,  
Q;MAAA,0BAAS,CAAT,I;MAAnB,OAAgB,CAAT,8BACgB,gBAAZ,qBAAK,KAAL,CAAY,CADhB,IAEoB,eA  
AhB,qBAAK,QAAQ,CAAR,IAAL,CAAgB,C;K;IAG/B,uC;MAGuD,ON3IyC,oBM2I/B,KAAM,MN3IyB,EM2IIB,  
KAAM,aAN,GAAqB,CAArB,IN3IkB,C;K;IM6IhG,yC;MAGqE,qCAAY,KAAM,MAAIB,EAAyB,KAAM,aAA  
N,GAAqB,CAArB,IAAzB,C;K;uFAErE,iC;MAS2E,2BAAY,KAAZ,EAAmB,GAAnB,C;K;mFAE3E,2C;MAO0D,  
wB;QAAA,WAAgB,gB;MAAkB,OAAA,8BAAY,UAAZ,EAAwB,QAAxB,CAAKC,W;K;IAE9H,uC;MAG6D,OA  
AA,8BAAY,KAAM,MAAIB,EAAyB,KAAM,aAN,GAAqB,CAArB,IAAzB,CAAiD,W;K;IAE9G,sE;MAImD,qC  
;QAAA,wBAAgC,S;MAC/E,YAAy,sBAAQ,SAAR,C;MACZ,OAAW,UAAS,EAAPB,GAAwB,qBAAxB,GNjL4F  
,oBMiL/B,CNjL+B,EMiL5B,KNjL4B,C;K;IMoLhG,wE;MAIqD,qC;QAAA,wBAAgC,S;MACjF,YAAy,sBAAQ,S  
AAR,C;MACZ,OAAW,UAAS,EAAPB,GAAwB,qBAAxB,GN1L4F,oBM0L/B,CN1L+B,EM0L5B,KN1L4B,C;K;I  
M6LhG,qE;MAIkD,qC;QAAA,wBAAgC,S;MAC9E,YAAy,sBAAQ,SAAR,C;MACZ,OAAW,UAAS,EAAPB,GA  
AwB,qBAAxB,GNnM4F,oBMmM/B,QAAQ,CAAR,INnM+B,EMmMpB,gBNnMoB,C;K;IMsMhG,uE;MAIoD,qC  
;QAAA,wBAAgC,S;MAChF,YAAy,sBAAQ,SAAR,C;MACZ,OAAW,UAAS,EAAPB,GAAwB,qBAAxB,GN5M4  
F,oBM4M/B,QAAQ,SAAU,OAAIB,IN5M+B,EM4ML,gBN5MK,C;K;IM+MhG,0E;MAIuD,qC;QAAA,wBAAgC,  
S;MACnF,YAAy,0BAAY,SAAZ,C;MACZ,OAAW,UAAS,EAAPB,GAAwB,qBAAxB,GNrN4F,oBMqN/B,CNrN  
+B,EMqN5B,KNrN4B,C;K;IMwNhG,4E;MAIyD,qC;QAAA,wBAAgC,S;MACrF,YAAy,0BAAY,SAAZ,C;MAC  
Z,OAAW,UAAS,EAAPB,GAAwB,qBAAxB,GN9N4F,oBM8N/B,CN9N+B,EM8N5B,KN9N4B,C;K;IMiOhG,yE;  
MAIsD,qC;QAAA,wBAAgC,S;MACIF,YAAy,0BAAY,SAAZ,C;MACZ,OAAW,UAAS,EAAPB,GAAwB,qBAAx  
B,GNvO4F,oBMuO/B,QAAQ,CAAR,INvO+B,EMuOpB,gBNvOoB,C;K;IM0OhG,2E;MAIwD,qC;QAAA,wBAAg  
C,S;MACpF,YAAy,0BAAY,SAAZ,C;MACZ,OAAW,UAAS,EAAPB,GAAwB,qBAAxB,GNhP4F,oBMgP/B,QA  
AQ,SAAU,OAAIB,INhP+B,EMgPL,gBNhPK,C;K;IMmPhG,oE;MAOI,IAAI,WAAW,UAAf,C;QACI,MAAM,8B  
AA0B,gBAAa,QAAb,oCAAkD,UAAID,OAA1B,C;MACV,SAAS,sB;MACT,EAAG,qBAAY,SAAZ,EAakB,CAA  
IB,EAAqB,UAArB,C;MACH,EAAG,gBAAO,WAAp,C;MACH,EAAG,qBAAY,SAAZ,EAakB,QAAIB,EAA4B,g  
BAA5B,C;MACH,OAAO,E;K;yFAGX,yB;MAAA,8B;MAAA,qD;MAAA,+D;QAOK,Q;QAAD,OAAuB,aAAtB,2  
DAAsB,EAAa,UAAb,EAAYB,QAazB,EAAmC,WAAnc,CAAgD,W;O;KAP3E,C;IASA,uD;MAOI,+BAAa,KAA  
M,MAAnB,EAA0B,KAAM,aAN,GAAqB,CAArB,IAA1B,EAakD,WAAID,C;K;yFAEJ,yB;MAAA,8B;MAAA,  
qD;MAAA,gD;QAOK,Q;QAAD,OAAuB,aAAtB,2DAAsB,EAAa,KAAb,EAAoB,WAApB,CAAiC,W;O;KAP5D,  
C;IASA,sD;MASI,IAAI,WAAW,UAAf,C;QACI,MAAM,8BAA0B,gBAAa,QAAb,oCAAkD,UAAID,OAA1B,C;M  
AEV,IAAI,aAAY,UAAhB,C;QACI,OAAy,mBAAL,SAAK,EAAY,CAAZ,EAAe,gBAAf,C;MAEHb,SAAS,mBA  
Ac,oBAAU,QAaV,GAAqB,UAArB,KAAd,C;MACT,EAAG,qBAAY,SAAZ,EAakB,CAAIB,EAAqB,UAArB,C;  
MACH,EAAG,qBAAY,SAAZ,EAakB,QAAIB,EAA4B,gBAA5B,C;MACH,OAAO,E;K;uFAGX,yB;MAAA,8B;  
MAAA,mD;MAAA,kD;QASK,Q;QAAD,OAAuB,YAAtB,2DAAsB,EAAY,UAAZ,EAAwB,QAAxB,CAAKC,W;O  
;KAT7D,C;IAWA,yC;MAKqE,8BAAY,KAAM,MAAIB,EAAyB,KAAM,aAN,GAAqB,CAArB,IAAzB,C;K;uFA  
ErE,yB;MAAA,8B;MAAA,mD;MAAA,mC;QAOK,Q;QAAD,OAAuB,YAAtB,2DAAsB,EAAY,KAaz,CAAmB,  
W;O;KAP9C,C;IASA,yC;MAKI,IAAI,wBAAW,MAAX,CAAJ,C;QACI,OAAO,8BAAY,MAAO,OAAhB,EAA2B,  
gBAA3B,C;;MAEX,OAAO,8BAAY,CAAZ,EAAe,gBAAf,C;K;IAGX,2C;MAKI,IAAI,wBAAW,MAAX,CAAJ,C;  
QACI,ONiWyE,oBMkWxD,MAAO,ONiWiD,C;;MMoW7E,OAAO,S;K;IAGX,yC;MAKI,IAAI,sBAAS,MAAT,C  
AAJ,C;QACI,OAAO,8BAAY,CAAZ,EAAe,mBAAS,MAAO,OAAhB,IAAf,C;;MAEX,OAAO,8BAAY,CAAZ,EA  
Ae,gBAAf,C;K;IAGX,2C;MAKI,IAAI,sBAAS,MAAT,CAAJ,C;QACI,ONrXwF,oBMqXvE,CNrxuE,EMqXpE,m  
BAAS,MAAO,OAAhB,INrXoE,C;;MMuX5F,OAAO,S;K;IAGX,sD;MAMI,IAAK,qBAAU,MAAO,OAAP,GAAg  
B,MAAO,OAAvB,IAAV,CAAD,IAA6C,wBAAW,MAAX,CAA7C,IAAmE,sBAAS,MAAT,CAAvE,C;QACI,OA  
AO,8BAAY,MAAO,OAAhB,EAA2B,mBAAS,MAAO,OAAhB,IAA3B,C;;MAEX,OAAO,8BAAY,CAAZ,EAAe,g  
BAAf,C;K;IAGX,wD;MAMI,IAAK,qBAAU,MAAO,OAAP,GAAgB,MAAO,OAAvB,IAAV,CAAD,IAA6C,wBA  
AW,MAAX,CAA7C,IAAmE,sBAAS,MAAT,CAAvE,C;QACI,ON7YwF,oBM6YvE,MAAO,ON7YgE,EM6YxD,  
mBAAS,MAAO,OAAhB,IN7YwD,C;;MM+Y5F,OAAO,S;K;IAGX,mD;MAKmF,oCAAKB,SAaIB,EAA6B,SAa7

B,C;K;IAEnF,mD;MAKuE,sCAAkB,SAAlB,EAA6B,SAA7B,C;K;IAEvE,iF;MAIsE,qC;QAAA,wBAAGC,S;MAC IG,YAAY,sBAAQ,SAAR,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QA5JvB,U;QA4JM,OA5 JgB,aAAtB,+DAAsB,EA4JyC,CA5JzC,EA4J4C,KA5J5C,EA4JmD,WA5JnD,CAAGD,W;;MA4JvE,W;K;IAGJ,mF; MAIwE,qC;QAAA,wBAAGC,S;MACpG,YAAY,sBAAQ,SAAR,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAA A,OAAiB,qB;;QArKvB,U;QAqKM,OArKgB,aAAtB,+DAAsB,EAqKyC,CARkzC,EAqK4C,KArK5C,EAqKmD,W ArKnD,CAAGD,W;;MAqKvE,W;K;IAGJ,gF;MAIqE,qC;QAAA,wBAAGC,S;MACjG,YAAY,sBAAQ,SAAR,C;M ACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QAA2B,iBAaA,QAAQ,CAAR,I;QAAb,eAAwB,gB;QA 9K1E,U;QA8KM,OA9KgB,aAAtB,+DAAsB,EAAa,UAAb,EAAYB,QAAzB,EA8K4D,WA9K5D,CAAGD,W;;MA 8KvE,W;K;IAGJ,kF;MAIuE,qC;QAAA,wBAAGC,S;MACnG,YAAY,sBAAQ,SAAR,C;MACL,Q;MAAA,IAAI,U AAS,EAAb,C;QAAA,OAAiB,qB;;QAA2B,iBAaA,QAAQ,SAAU,OAAIB,I;QAAb,eAAuC,gB;QAvLzF,U;QAuL M,OAvLgB,aAAtB,+DAAsB,EAAa,UAAb,EAAYB,QAAzB,EAuL2E,WAvL3E,CAAGD,W;;MAuLvE,W;K;IAGJ, oF;MAI2E,qC;QAAA,wBAAGC,S;MACvG,YAAY,0BAAY,SAAZ,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QA AA,OAAiB,qB;;QAA2B,iBAaA,QAAQ,SAAU,OAAIB,I;QAAb,eAAuC,gB;QAhMzF,U;QAgMM,OAHMgB,aAAt B,+DAAsB,EAAa,UAAb,EAAYB,QAAzB,EAgM2E,WAhM3E,CAAGD,W;;MAGMvE,W;K;IAGJ,sF;MAIyE,qC; QAAA,wBAAGC,S;MACrG,YAAY,0BAAY,SAAZ,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB ;;QAA2B,iBAaA,QAAQ,CAAR,I;QAAb,eAAwB,gB;QAzM1E,U;QAYMM,OAzMgB,aAAtB,+DAAsB,EAAa,UA Ab,EAAYB,QAAzB,EAyM4D,WzM5D,CAAGD,W;;MAyMvE,W;K;IAGJ,qF;MAI0E,qC;QAAA,wBAAGC,S;M ACtG,YAAY,0BAAY,SAAZ,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QAlNvB,U;QakNM, OAlNgB,aAAtB,+DAAsB,EakNyC,CAINzC,EakN4C,KAIN5C,EakNmD,WAlNnD,CAAGD,W;;MAkNvE,W;K; IAGJ,uF;MAI4E,qC;QAAA,wBAAGC,S;MACxG,YAAY,0BAAY,SAAZ,C;MACL,Q;MAAA,IAAI,UAAS,EAAb, C;QAAA,OAAiB,qB;;QA3NvB,U;QA2NM,OA3NgB,aAAtB,+DAAsB,EA2NyC,CA3NzC,EA2N4C,KA3N5C,EA 2NmD,WA3NnD,CAAGD,W;;MA2NvE,W;K;+EAOJ,yC;MAQoF,OAAA,KAAM,iBAAQ,SAAR,EAAC,WAAd,C; K;+EAE1F,uC;MAOI,OAAA,KAAM,iBAAQ,SAAR,EAAC,SAAd,C;K;yFAEV,yC;MAMyF,OAAA,KAAM,sBA Aa,SAAb,EAAMb,WAAAnB,C;K;+FAE/F,yB;MAAA,oC;MAAA,gC;MAAA,uC;QAeW,Q;QAAA,IApe4C,mBAA S,CAoerD,C;uBAAkB,oBAAU,iCAAK,CAAL,EAAY,E;UAAA,YNzhBoD,oBMyhBrB,CNzhBqB,C;UMyhBtE,O LrjBwD,2BAAL,GAakB,K;;UKqjBrE,OAAyD,S;QAAhE,W;O;KafJ,C;iGakBA,yB;MAAA,oC;MAAA,uC;QAEI ,OatfmD,mBAAS,CAsf5D,GAAyB,UAAU,iCAAK,CAAL,EAAY,CAAmB,WAAAnB,GN3iBoD,oBM2iBV,CN3iB U,CM2iB7E,GAA2E,S;O;Kaf/E,C;+EAmBA,4B;MAIsE,OAAA,KAAM,iBAAQ,SAAR,C;K;IAE5E,0F;MAKI,IA AK,cAAc,CAAf,IAAsB,aAAa,CAAnC,IAA0C,cAAa,SAAK,OAAL,GAAC,MAAd,IAAb,CAA1C,IAAiF,eAAc,K AAM,OAAN,GAAe,MAAf,IAAd,CAArF,C;QACI,OAAO,K;;MAGX,iBAAC,CAAd,UAAAsB,MAAtB,U;QACI,IA AI,CAA0B,SAAZB,qBAAK,aAAa,KAAb,IAAL,CAAYB,EAAO,iBAAM,cAAc,KAAd,IAAN,CAAP,EAAMC,UA AnC,CAA9B,C;UACI,OAAO,K;;MAEf,OAAO,I;K;IAGX,mD;MAG+C,0B;QAAA,aAsB,K;MACjE,OAAA,SA AK,OAAL,GAAC,CAAd,IAA2B,SAAR,qBAAK,CAAL,CAAQ,EAAO,IAAP,EAAa,UAAb,C;K;IAE/B,iD;MAG6 C,0B;QAAA,aAsB,K;MAC/D,OAAA,SAAK,OAAL,GAAC,CAAd,IAAmC,SAAhB,qBAAK,2BAAL,CAAGB,E AAO,IAAP,EAAa,UAAb,C;K;IAEvC,qD;MAGyD,0B;QAAA,aAsB,K;MAC3E,IAAI,CAAC,UAAD,IAAe,6BA Af,IAAiC,0BAArC,C;QACI,OAAy,WAAL,SAAK,EAAY,MAAX,C;;QAEZ,OAAO,6BAAkB,CAAlB,EAqB,M AArB,EAA6B,CAA7B,EAAGC,MAAO,OAAvC,EAA+C,UAA/C,C;K;IAGf,iE;MAG0E,0B;QAAA,aAsB,K;MA C5F,IAAI,CAAC,UAAD,IAAe,6BAAf,IAAiC,0BAArC,C;QACI,OAAy,aAAL,SAAK,EAAY,MAAX,EAAMb,U AAnB,C;;QAEZ,OAAO,6BAAkB,UAAIB,EAA8B,MAA9B,EAAsC,CAAtC,EAAYC,MAAO,OAAhD,EAAD,U AAxD,C;K;IAGf,mD;MAGuD,0B;QAAA,aAsB,K;MACzE,IAAI,CAAC,UAAD,IAAe,6BAAf,IAAiC,0BAArC, C;QACI,OAAy,SAAL,SAAK,EAAS,MAAT,C;;QAEZ,OAAO,6BAAkB,mBAAS,MAAO,OAAhB,IAAIB,EAA0C ,MAA1C,EAakD,CAAlD,EAqD,MAAO,OAA5D,EAAoE,UAApE,C;K;IAMf,wD;MAQ8D,0B;QAAA,aAsB,K ;MACHf,qBfjnBO,MAAO,KeinBa,SAAK,OfjnBIB,EeinB0B,KAAM,OfjnBhC,C;MemnBd,QAAQ,C;MACR,OAA O,IAAI,cAAJ,IAA8B,SAAR,qBAAK,CAAL,CAAQ,EAAO,iBAAM,CAAN,CAAP,EA8B,UAA9B,CAArC,C;Q ACI,a;;MAEJ,IAAS,mBAAL,SAAK,EAAMb,IAAI,CAAJ,IAAnB,CAAL,IAAwC,mBAAN,KAAM,EAAMb,IAAI ,CAAJ,IAAnB,CAA5C,C;QACI,a;;MAEJ,OAAO,8BAAY,CAAZ,EAae,CAAF,CAAKB,W;K;IAG7B,wD;MAQ8D ,0B;QAAA,aAsB,K;MACHf,iBAAiB,SAAK,O;MACtB,kBAakB,KAAM,O;MACxB,qBfxoBO,MAAO,KewoBa ,UfxoBb,EewoByB,WfxoBzB,C;Me0oBd,QAAQ,C;MACR,OAAO,IAAI,cAAJ,IAA+C,SAAZB,qBAAK,aAAa,CA

Ab,GAAiB,CAAjB,IAAL,CAAyB,EAAO,iBAAM,cAAc,CAAd,GAakB,CAAIB,IAAN,CAAP,EAAGd,UAAhD,C  
AAtD,C;QACI,a;;MAEJ,IAAS,mBAAL,SAAK,EAAMb,AAaA,CAAb,GAAiB,CAAjB,IAAnB,CAAL,IAAQd,mB  
AAN,KAAM,EAAMb,cAAc,CAAd,GAakB,CAAIB,IAAnB,CAAzD,C;QACI,a;;MAEJ,OAAO,8BAAy,AAaA,CA  
Ab,IAAZ,EAA4B,UAA5B,CAAwC,W;K;IAMnD,8D;MAQqD,0B;QAAA,aAAkB,C;MAAG,0B;QAAA,aAAsB,K;  
MAMnE,UAAkB,M;MAL3C,IAAI,CAAC,UAAD,IAAe,KAAM,OAAN,KAAC,CAA7B,IAAkC,6BAAAtC,C;QACI,  
WAAiB,SAAN,KAAM,C;QACjB,ONjtBwF,kB0G3ME,oBpG45BrE,IoG55BqE,C1G2MF,EMitB7D,UNjtB6D,C;;  
MMotBnE,uBAAX,UAAW,EAAC,CAAd,C;MAAkB,oC;kBAA3C,gD;QACI,kBAAkB,qBAAI,KAAJ,C;QACR,c;;  
UICwkXE,U;UAAhB,4BkCkXQ,KICwkXR,kB;YAAgB,cAAhB,UkCkXQ,KICwkXR,S;YAAsB,IkCkXC,SAA  
H,UICwkXgB,oBkCkXhB,CAAG,0BICwkXD,C;cAAwB,AAAO,I;cAAP,e;;;UAC9C,AAAO,K;;;QkCzkXH,e;UA  
CI,OAAO,K;;MAEf,OAAO,E;K;IAGX,KE;MASyD,0B;QAAA,aAAkB,2B;MAAW,0B;QAAA,aAAsB,K;MACxG,  
IAAI,CAAC,UAAD,IAAe,KAAM,OAAN,KAAC,CAA7B,IAAkC,6BAAAtC,C;QACI,WAAiB,SAAN,KAAM,C;QA  
CjB,ONruB4F,sB0G3MM,oBpGg7BzE,IoGh7ByE,C1G2MN,EMquB7D,UNruB6D,C;;kBMyuBhG,iBAAYB,eAA  
X,UAAW,EAaA,2BAAb,CAAzB,WAAwD,CAAxD,U;QACI,kBAAkB,qBAAI,KAAJ,C;QACR,c;;UICgjXE,Q;UA  
AhB,wBkChjXQ,KICgjXR,gB;YAAgB,cAAhB,UkChjXQ,KICgjXR,O;YAAsB,IkChjXC,SAAH,UICgjXgB,oBkC  
hjXhB,CAAG,0BICgjXD,C;cAAwB,AAAO,I;cAAP,e;;;UAC9C,AAAO,K;;;QkCjjXH,e;UACI,OAAO,K;;MAGf,OA  
AO,E;K;IAIX,8E;MAA2G,oB;QAAA,OAAGB,K;MAOrG,UAKA,M;MAXIB,cAAkB,CAAC,IAAL,GACV,aAAW  
,gBAAX,UAAW,EAAC,CAAd,CAAX,EAAsC,eAAT,QAAS,EAaA,gBAAb,CAAtC,CADU,GAGV,SAAW,eAAX,  
UAAW,EAaA,2BAAb,CAAX,EAAMd,gBAAT,QAAS,EAAC,CAAd,CAAnD,C;MAEJ,IAAI,iCAAkB,yBAAtB,C;  
QACkB,yB;QAAd,OAAC,CAAd,C;UAAc,uB;UACV,IAAU,cAAN,KAAM,EAAC,CAAd,EAiB,SAAJB,EAABuB,  
KAAvB,EAa8B,KAAM,OAAPc,EAa4C,UAA5C,CAAV,C;YACI,OAAO,K;;;QAGD,2B;QAAd,OAAC,gBAAd,  
C;UAAc,2B;UACV,IAAU,kBAAN,KAAM,EAakB,CAAIB,EAaqB,SAArB,EAa2B,OAA3B,EAakC,KAAM,OA  
AxC,EAAGd,UAAhD,CAAV,C;YACI,OAAO,O;;;MAGnB,OAAO,E;K;IAGX,qE;MAUsB,UAMA,M;MAfIB,IAA  
I,CAAC,UAAD,IAAe,OAAQ,KAAR,KAAGB,CAAnC,C;QACI,AAaQb,UAR,OAAQ,C;QACrB,YAAgB,CAAC,  
IAAL,GAAW,sBAAQ,MAAR,EAAGB,UAAhB,CAAX,GAA4C,0BAAy,MAAZ,EAaOB,UAApB,C;QACxD,OA  
AW,QAAQ,CAAZ,GAAe,IAAf,GAAYB,UAAAS,MAAT,C;;MAGpC,cAAkB,CAAC,IAAL,GAAW,aAAW,gBAAX  
,UAAW,EAAC,CAAd,CAAX,EAa6B,gBAA7B,CAAX,GAAoD,SAAW,eAAX,UAAW,EAaA,2BAAb,CAAX,EA  
A0C,CAA1C,C;MAEIE,IAAI,6BAAJ,C;QACkB,yB;oBAAd,OAAC,CAAd,C;UAAc,yB;UACmB,sB;;Yb3sBrB,U;Y  
AAA,Sa2sBa,Ob3sBb,W;YAAhB,OAAGB,gBAAhB,C;cAAGB,2B;cAAM,Ia2sBgC,Cb3sBIB,Oa2sBkB,EAAC,CAA  
d,sBb3sBIB,Oa2sBmD,OAAjC,ab3sBhC,C;gBAAwB,qBAAO,O;gBAAP,uB;;;YAC9C,qBAAO,I;;;Ua0sBC,uC;U  
ACA,IAAI,sBAAJ,C;YACI,OAAO,YAAS,cAAT,C;;;QAGD,2B;oBAAd,OAAC,gBAAd,C;UAAc,2B;UACmB,wB;  
;YbjtBrB,U;YAAA,SaitBa,ObjtBb,W;YAAhB,OAAGB,gBAAhB,C;cAAGB,6B;cAAM,IaitBgC,kBbjtBIB,SaitBkB,  
EAakB,CAAIB,sBbjtBIB,SaitBuD,OAARc,abjtBhC,C;gBAAwB,uBAAO,S;gBAAP,uB;;;YAC9C,uBAAO,I;;;Uagt  
BC,2C;UACA,IAAI,wBAAJ,C;YACI,OAAO,YAAS,gBAAT,C;;;MAInB,OAAO,I;K;IAGX,iE;MAY+D,0B;QAA  
A,aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MACtG,4BAAU,OAaV,EAAMb,UAAAnB,EAa+B,UAA/B,EAakD,KA  
AID,C;K;IAEJ,mE;MAYmE,0B;QAAA,aAAkB,2B;MAAW,0B;QAAA,aAAsB,K;MACIH,4BAAU,OAaV,EAAM  
B,UAAAnB,EAa+B,UAA/B,EAakD,IAAID,C;K;IAEJ,KE;MAWgE,0B;QAAA,aAAkB,C;MAAG,0B;QAAA,aAAs  
B,K;MACvG,gB;MAAA,8CAAU,OAaV,EAAMb,UAAAnB,EAa+B,UAA/B,EAakD,KAAID,mDAAMe,E;K;IAE  
vE,sE;MAYoE,0B;QAAA,aAAkB,2B;MAAW,0B;QAAA,aAAsB,K;MACnH,gB;MAAA,8CAAU,OAaV,EAAMb  
,UAAAnB,EAa+B,UAA/B,EAakD,IAAID,mDAaKE,E;K;IAKtE,6D;MAM4C,0B;QAAA,aAAkB,C;MAAG,0B;Q  
AAA,aAAsB,K;MACnF,OAaW,cAAc,gCAAZB,GACI,sBAAW,mBAAY,IAAZ,CAAX,EAa8B,UAA9B,EAa0C,  
UAA1C,CADJ,GNz2B4F,kB0G3ME,oBpGujC5E,IoGvjC4E,C1G2MF,EM42BpE,UN52BoE,C;K;IM+2BhG,+D;M  
AQgD,0B;QAAA,aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MACvF,OAaW,cAAc,gCAAZB,GACI,sBAAQ,MAAR,  
EAAGB,UAAhB,EAa4B,gBAA5B,EAaOC,UAApC,CADJ,GNx3B4F,kBM23B1E,MN33B0E,EM23BIE,UN33Bk  
E,C;K;IM83BhG,iE;MAQgD,0B;QAAA,aAAkB,2B;MAAW,0B;QAAA,aAAsB,K;MAC/F,OAaW,cAAc,gCAAZ  
B,GACI,0BAaE,mBAAY,IAAZ,CAAf,EAakC,UAAIC,EAa8C,UAA9C,CADJ,GNp4BgG,sB0G3MM,oBpGklCh  
F,IoGllCgF,C1G2MN,EMu4BpE,UNv4BoE,C;K;IM04BpG,mE;MAQoD,0B;QAAA,aAAkB,2B;MAAW,0B;QAA  
A,aAAsB,K;MACnG,OAaW,cAAc,gCAAZB,GACI,sBAAQ,MAAR,EAAGB,UAAhB,EAa4B,CAA5B,EAa+B,U  
AA/B,EAakD,IAAID,CADJ,GNn5BgG,sBMs5B1E,MNt5B0E,EMs5BIE,UNt5BkE,C;K;IMy5BpG,mD;MAM+D,

0B;QAAA,aAAsB,K;MACjF,OAAI,yBAAJ,GACI,sBAAQ,KAAR,UAA4B,UAA5B,KAA2C,CAD/C,GAGI,sBAA  
Q,KAAR,EAAe,CAAf,EAakB,gBAAIB,EAA0B,UAA1B,KAAY,C,C;K;IAIjD,kD;MAMsD,0B;QAAA,aAAsB,K;  
MACxE,6BAAQ,IAAR,UAA2B,UAA3B,KAA0C,C;K;kFAE9C,4B;MAI0E,OAAA,KAAM,yBAAgB,SAAhB,C;K  
;IAM3C,yE;MACjC,oB;MACA,8B;MACA,oB;MACA,kC;K;IAG8C,sF;MAAA,gE;MAC1C,iBAAqB,E;MACrB,y  
BAAwC,WAAx,yCAAW,EAAS,CAAT,EAAY,oCAAM,OAAIB,C;MACxC,uBAA2B,sB;MAC3B,gBAA0B,I;M  
AC1B,eAAmB,C;K;0EAEnB,Y;MACI,IAAI,uBAAkB,CAAtB,C;QACI,iBAAY,C;QACZ,gBAAW,I;QAEX,IAAI,  
4CAAQ,CAAR,IAAa,uDAAa,yCAA1B,IAAmC,uBAAkB,yCAAM,OAA/D,C;UACI,gBAAW,qCAAyB,iBAAN,y  
CAAM,CAAzB,C;UACX,uBAAkB,E;;UAEIB,YAAkB,iDAAN,yCAAM,EAAa,oBAAb,C;UACIB,IAAI,SAAS,IA  
Ab,C;YACI,gBAAW,qCAAyB,iBAAN,yCAAM,CAAzB,C;YACX,uBAAkB,E;;YAEIB,IAAK,QAAiB,KAAjB,aA  
AL,EAAY,SAAU,KAAV,a;YACZ,gBAAW,gCAAwB,KAAxB,C;YACX,yBAAoB,QAAQ,MAAR,I;YACpB,uBAA  
AkB,0BAAwB,WAAU,CAAd,GAAiB,CAAjB,GAAwB,CAA5C,K;;;QAG1B,iBAAY,C;;K;oEAIpB,Y;MAKiB,Q;  
MAJb,IAAI,mBAAa,EAAjB,C;QACI,iB;MACJ,IAAI,mBAAa,CAAjB,C;QACI,MAAM,6B;MACV,aAAa,mE;MA  
Eb,gBAAW,I;MACX,iBAAY,E;MACZ,OAAO,M;K;uEAGX,Y;MACI,IAAI,mBAAa,EAAjB,C;QACI,iB;MACJ,  
OAAO,mBAAa,C;K;;iDA9C5B,Y;MAA8C,+D;K;;IAGeU,0E;MAAA,0C;QhB1mCjD,SgB2mCH,sBAAW,kBAA  
X,EAAuB,YAAvB,EAakD,kBAAID,C;QAAA,OAAwE,KAAK,CAAT,GAAY,IAAZ,GAA5B,OAAM,CAAN,C;O  
;K;IAdIG,iF;MAUkE,0B;QAAA,aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MAAO,qB;QAAA,QAAa,C;MAC7H,wB  
AAwB,KAAxB,C;MAEA,OAAO,4BAAwB,SAAXB,EAA8B,UAA9B,EAA0C,KAA1C,EAAiD,gDAAjD,C;K;IAw  
BiD,gF;MAAA,0C;QAAkB,Q;QAAA,oCAAU,sBAAV,EAA0B,YAA1B,EAAqD,kBAArD,EAAwE,KAAxE,aAAs  
F,GAAG,UAAH,EAAe,WAAO,OAAtB,CAAtF,O;O;K;IAIB9E,mF;MAc0E,0B;QAAA,aAAkB,C;MAAG,0B;QA  
AA,aAAsB,K;MAAO,qB;QAAA,QAAa,C;MACrI,wBAAwB,KAAxB,C;MACA,qBAAgC,OAAX,UAAW,C;MAE  
hC,OAAO,4BAAwB,SAAXB,EAA8B,UAA9B,EAA0C,KAA1C,EAAiD,sDAAjD,C;K;IAIX,wC;MnBltCI,IAAI,E  
mBmtCI,SAAS,CnBntCb,CAAJ,C;QACI,cmBktCkB,8C;QnBjtCIB,MAAM,gCAAyB,OAAQ,WAAjC,C;;K;ImBku  
CgE,sD;MAAA,qB;QAAE,yCAAU,EAAV,C;O;K;IAZhF,mE;MAWmE,0B;QAAA,aAAsB,K;MAAO,qB;QAAA,  
QAAa,C;MACzG,OAAsE,OAAtE,+BAakB,UAAIB,UAA2C,UAA3C,EAA+D,KAA/D,CAAsE,EAAL,iCAAJ,C;K  
;IAE1E,yD;MAWyD,0B;QAAA,aAAsB,K;MAAO,qB;QAAA,QAAa,C;MAC/F,IAAI,UAAW,OAAX,KAAmB,CA  
AvB,C;QACI,gBAAgB,WAAW,CAAX,C;QACHB,IAAI,EAAC,SAh/BuC,YAAU,CAG/BID,CAAJ,C;UACI,OAA  
O,mBAAM,SAAN,EAAiB,UAAjB,EAA6B,KAA7B,C;;;MAI2E,kBAAb,cAAAtE,+BAakB,UAAIB,UAA2C,UAA3  
C,EAA+D,KAA/D,CAAsE,C;MbgPtE,kBAAM,iBAAa,qCAAwB,EAAxB,CAAb,C;MAuEA,Q;MAAA,6B;MAAb,  
OAAa,cAAb,C;QAAa,sB;QACT,WAAy,WaxTgF,uBbwTIE,IaxTkE,CbwThF,C;;MaxThB,ObyTO,W;K;Ia9SmE,w  
D;MAAA,qB;QAAE,yCAAU,EAAV,C;O;K;IARhF,qE;MAOiE,0B;QAAA,aAAsB,K;MAAO,qB;QAAA,QAAa,C;  
MACvG,OAAsE,OAAtE,6BAakB,UAAIB,UAA2C,UAA3C,EAA+D,KAA/D,CAAsE,EAAL,mCAAJ,C;K;IAE1E,  
2D;MAOuD,0B;QAAA,aAAsB,K;MAAO,qB;QAAA,QAAa,C;MAC7F,IAAI,UAAW,OAAX,KAAmB,CAAvB,C;  
QACI,OAAO,mBAAoB,oBAAAd,WAAW,CAAX,CAAc,CAApB,EAAGC,UAAhC,EAA4C,KAA5C,C;;MAG+E,kB  
AAb,cAAAtE,6BAakB,UAAIB,UAA2C,UAA3C,EAA+D,KAA/D,CAAsE,C;MbuNtE,kBAAM,iBAAa,qCAAwB,E  
AAxB,CAAb,C;MAuEA,Q;MAAA,6B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAy,Wa/RgF,uBb+RIE,Ia/RkE  
,Cb+RhF,C;;Ma/RhB,ObgSO,W;K;Ia7RX,0D;MASI,wBAAwB,KAAxB,C;MAEA,oBAAoB,C;MACpB,gBAAgB,s  
BAAQ,SAAR,EAAMB,aAAnB,EAakC,UAAIC,C;MACHB,IAAI,cAAa,EAAb,IAAmB,UAA5C,CAAhC,C;QACI,O  
AAO,OAAO,SAAK,WAAZ,C;;MAGX,gBAAgB,QAAQ,C;MACxB,aAAa,iBAAsB,SAAJ,GAAqB,eAAN,KAAM,  
EAAa,EAAb,CAArB,GAA2C,EAA7D,C;;QAET,MAAO,WA36B6E,8BA26B/D,aA36B+D,EA26BhD,SA36BgD,  
CAAkC,WA26B/G,C;QACP,gBAAgB,YAAy,SAAU,OAAtB,I;QAEhB,IAAI,aAAa,MAAO,KAAP,MAAE,QAAQ  
,CAAR,IAAf,CAAjB,C;UAA2C,K;QAC3C,YAAy,sBAAQ,SAAR,EAAMB,aAAnB,EAakC,UAAIC,C;;MACP,sB  
AAa,EAAb,C;MAET,MAAO,WAI7BiF,8BAk7BnE,aAI7BmE,EAK7BpD,gBAI7BoD,CAAKC,WAK7BnH,C;MAC  
P,OAAO,M;K;2EAGX,mC;MAOmD,qB;QAAA,QAAa,C;MAAmB,OAAA,KAAM,eAAM,SAAN,EAAY,KAAZ,  
C;K;+FAEzF,mC;MAU6D,qB;QAAA,QAAa,C;MAAuB,OAAA,KAAM,yBAAgB,SAAhB,EAAsB,KAAtB,C;K;I  
AEvG,iC;MAK2D,mCAAgB,MAAhB,EAAwB,IAAxB,EAA8B,IAA9B,E;K;IAE3D,0B;MAKgD,OAAe,UAAf,uB  
AAe,C;K;IAqB/D,uD;MAQsB,Q;MAPIB,IAAI,iCAAkB,yBAAtB,C;QACI,OAAy,SAAL,SAAK,EAAO,KAAP,E  
AA2B,IAA3B,C;;MAGhB,IAAI,cAAS,KAAb,C;QAAoB,OAAO,I;MAC3B,IAAI,qBAAgB,aAAhB,IAAiC,SAAK,  
OAAAL,KAAe,KAAM,OAA1D,C;QAAkE,OAAO,K;MAEvD,uB;MAAIB,aAAU,CAAV,gB;QACI,IAAI,CAAS,SA

AR,qBAAK,CAAL,CAAQ,EAAO,iBAAM,CAAN,CAAP,EAA8B,IAA9B,CAAb,C;UACI,OAAO,K;;;MAIf,OAA  
O,I;K;IAGX,6C;MAQsB,Q;MAPIB,IAAI,iCAAkB,yBAAtB,C;QACI,OAAO,kBAAQ,KAAR,C;;MAGX,IAAI,cA  
AS,KAAb,C;QAAoB,OAAO,I;MAC3B,IAAI,qBAAgB,aAAhB,IAAiC,SAAK,OAAAL,KAAe,KAAM,OAA1D,C;Q  
AAkE,OAAO,K;MAEvD,uB;MAAIB,aAAU,CAAV,gB;QACI,IAAI,qBAAK,CAAL,MAAW,iBAAM,CAAN,CA  
Af,C;UACI,OAAO,K;;;MAIf,OAAO,I;K;IAGX,oC;MAU+C,QAAM,SAAN,C;aAC3C,M;UAD2C,OACjC,I;aACV  
,O;UAF2C,OAEhC,K;;UACH,MAAM,gCAAYB,mDAAGD,SAAzE,C;;K;IAGIB,0C;MAUsD,QAAM,SAAN,C;aA  
CID,M;UADkD,OACxC,I;aACV,O;UAFkD,OAEvC,K;;UAFuC,OAG1C,I;;K;IgLr8CZ,sB;MAAA,0B;MAII,aAC+  
B,e;MAC/B,cACgC,e;MACHc,WAC6B,e;MAC7B,YAC8B,e;MAC9B,eACiC,e;MACjC,YAC8B,gB;MAC9B,aAC  
+B,gB;MAC/B,YAC8B,gB;MAC9B,aAC+B,gB;MAC/B,eACiC,gB;MACjC,iBACmC,gB;MACnC,qBAEuC,gB;M  
ACvC,sBAEwC,gB;MACxC,kBACoC,gB;MACpC,cACgC,gB;MACHc,iBACmC,gB;MACnC,iBACmC,gB;MAC  
nC,iBACmC,gB;MACnC,YAC8B,gB;MAC9B,aAC+B,iB;MAC/B,aAC+B,iB;MAC/B,uBACyC,iB;MACzC,wBA  
C0C,iB;MAC1C,sBACwC,iB;MACxC,uBACyC,iB;MACzC,wBAC0C,iB;MAC1C,sBACwC,iB;MACxC,cACgC,i  
B;MACHc,oBACsC,iB;MACTc,cACgC,iB;MACHc,gBACkC,iB;MACiC,aAC+B,iB;MAC/B,mBACqC,iB;MACrC  
,YAC8B,iB;MAC9B,UAC4B,iB;MAC5B,mBACqC,iB;MACrC,gBACkC,iB;MACiC,mBACqC,iB;MACrC,sBAC  
wC,iB;MAExC,sBAGwC,gB;MAExC,uBAGyC,gB;K;;;IA7F7C,kC;MAAA,iC;QAAA,gB;;MAAA,0B;K;;;;2F  
CwE0C,Y;MAAQ,oCAAa,IAAb,C;K;IAiBpB,yC;MAAqB,kB;K;mIAC3C,Y;MACmD,OAAA,UAAM,YAAN,aA  
AkB,CAAIB,C;K;mIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CAAIB,C;K;mIACnD,Y;MACmD,OAAA,  
UAAM,YAAN,aAAkB,CAAIB,C;K;mIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CAAIB,C;K;mIACnD,  
Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CAAIB,C;K;mIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CA  
AIB,C;K;mIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CAAIB,C;K;mIACnD,Y;MACmD,OAAA,UAAM,  
YAAN,aAAkB,CAAIB,C;K;mIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CAAIB,C;K;qIACnD,Y;MACm  
D,OAAA,UAAM,YAAN,aAAkB,EAAIB,C;K;gDAEnD,Y;MAMoC,OAAA,UAAM,YAAY,iBAAQ,CAAR,EAA  
W,UAAM,YAAY,KAA7B,C;K;;;6ErEIH9D,yB;MAAA,iD;MAAA,4B;QAI4C,kBAAM,SAAN,C;O;KAJ5C,C;+E  
AMA,yB;MAAA,gD;MAAA,oC;QAI+D,kBAAM,SAAN,EAA Y,MAAZ,C;O;KAJ/D,C;+EAMA,yB;MAAA,oC;M  
AAA,qC;QAIqE,sBAAM,SAAN,EAA Y,OAAZ,C;O;KAJrE,C;IplY4B,4B;MAmBxB,gC;MAnB6C,0B;MAW7B,U  
AEA,MAFA,EAGA,M;MALZ,I+HjC8D,I/Hic9D,C;QACI,IAAI,kBAAJ,C;UACQ,mB;UAAJ,IAAI,sEAA sB,SAAt  
B,EAAJ,C;YAAqC,MAAM,sBAAiB,YAAF,+CAAf,C;;UAEvC,qB;UAAJ,IAAI,0EAAuB,UAAvB,EAAJ,C;YAAu  
C,MAAM,sBAAiB,YAAF,gDAAf,C;UACzC,qB;UAAJ,IAAI,kEAA+B,mBAA/B,CAAJ,C;YAAwD,MAAM,sBA  
AiB,YAAF,mCAAf,C;;K;mFAZID,Y;MAAQ,kCAAa,CAAb,C;K;+FACU,Y;MAAQ,OAAA,eAAS,QAAT,GAAq  
B,C;K;qCACvE,Y;MAA0B,QADwB,eAAS,QAAT,GAAqB,CAC7C,MAAqB,C;K;sCAC/C,Y;MAA2B,QAFuB,eA  
AS,QAAT,GAAqB,CAE5C,MAAqB,C;K;yFACxB,Y;MAAQ,OAAI,kBAAJ,mF;K;IAahC,8B;MAAA,kC;MACI,Y  
AC4B,gB;MAE5B,gBACgC,iBAAiB,UAAjB,C;MACHc,4BAAsC,uC;K;mDAEtC,yC;MAGI,2BAAoB,KAAPB,E  
AA2B,UAA3B,EAAuC,UAAvC,C;K;iJAM8B,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,sD;O;KAAR,C;iJAIC  
,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,sD;O;KAAR,C;iJAUE,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ  
,sD;O;KAAR,C;mJAKF,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,uD;O;KAAR,C;mJAIC,yB;MAAA,6C;MA  
AA,iD;MAAA,4B;QAAQ,uD;O;KAAR,C;mJAUE,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,uD;O;KAAR,C;  
mJAKH,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,uD;O;KAAR,C;mJAIC,yB;MAAA,6C;MAAA,iD;MAAA,4  
B;QAAQ,uD;O;KAAR,C;mJAUE,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,uD;O;KAAR,C;yIAKR,yB;MAA  
A,6C;MAAA,iD;MAAA,4B;QAAQ,kD;O;KAAR,C;yIAIC,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,kD;O;K  
AAR,C;yIAUE,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,kD;O;KAAR,C;yIAKH,yB;MAAA,6C;MAAA,iD;M  
AAA,4B;QAAQ,kD;O;KAAR,C;yIAIC,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,kD;O;KAAR,C;yIAUE,yB;  
MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,kD;O;KAAR,C;qIAKL,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,gD  
;O;KAAR,C;qIAIC,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,gD;O;KAAR,C;qIAUE,yB;MAAA,6C;MAAA,i  
D;MAAA,4B;QAAQ,gD;O;KAAR,C;mIAKJ,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,+C;O;KAAR,C;mIAIC  
,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,+C;O;KAAR,C;mIAUE,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAA  
Q,+C;O;KAAR,C;uDAK9B,iB;MAK+C,OAAM,WAAN,KAAM,yC;K;uDAErD,iB;MAKgD,OAAM,aAAN,KAA  
M,yC;K;uDAEtD,iB;MAskD,OAAM,aAAN,KAAM,yC;K;wDAGxD,iB;MAKgD,OAAM,WAAN,KAAM,0C;K;w  
DAEtD,iB;MAKiD,OAAM,aAAN,KAAM,0C;K;wDAEvD,iB;MASmD,OAAM,aAAN,KAAM,0C;K;wDAGzD,iB



;MAKgD,OAAM,WAAN,KAAM,0C;K;wDAEtD,iB;MAKiD,OAAM,aAAN,KAAM,0C;K;wDAEvD,iB;MASmD,OAAM,aAAN,KAAM,0C;K;mDAGzD,iB;MAK2C,OAAM,WAAN,KAAM,qC;K;mDAEjD,iB;MAK4C,OAAM,aAAN,KAAM,qC;K;mDAEID,iB;MAS8C,OAAM,aAAN,KAAM,qC;K;mDAGpD,iB;MAK2C,OAAM,WAAN,KAAM,qC;K;mDAEjD,iB;MAK4C,OAAM,aAAN,KAAM,qC;K;mDAEID,iB;MAS8C,OAAM,aAAN,KAAM,qC;K;iDAGpD,iB;MAKyC,OAAM,WAAN,KAAM,mC;K;iDAE/C,iB;MAK0C,OAAM,aAAN,KAAM,mC;K;iDAEhD,iB;MAS4C,OAAM,aAAN,KAAM,mC;K;gDAGID,iB;MAKwC,OAAM,WAAN,KAAM,kC;K;gDAE9C,iB;MAKyC,OAAM,aAAN,KAAM,kC;K;gDAE/C,iB;MAS2C,OAAM,aAAN,KAAM,kC;K;iDAEjD,iB;;QAY4C,OACxC,cAAc,KAAd,EAAiC,KAAjC,C;;QACF,+C;UACE,MAAM,6BAAYB,sCAAmC,KAAnC,OAazB,EAASe,CAATe,C;;UAHkC,O;;K;0DAM5C,iB;;QAeqD,OACjD,cAAc,KAAd,EAAiC,IAAjC,C;;QACF,+C;UACE,MAAM,6BAAYB,0CAAuC,KAAvC,OAazB,EAA0E,CAA1E,C;;UAH2C,O;;K;uDAMrD,iB;;QAWmD,OAC/C,cAAc,KAAd,EAAiC,KAAjC,C;;QACF,+C;UAFiD,OAG/C,I;;UAH+C,O;;K;gEAMnD,iB;;QAO4D,OACxD,cAAc,KAAd,EAAiC,IAAjC,C;;QACF,+C;UAF0D,OAGxD,I;;UAHwD,O;;K;;;IA1YhE,0C;MAAA,yC;QAAA,wB;;MAAA,kC;K;oCAmZA,Y;MAC6C,kBAAY,YAAD,aAX,EApA,kEaAS,QAAT,GAAqB,CAoa1B,C;K;qCAE7C,iB;MAiBW,Q;MATH,IAAA,IAAK,aAAL,C;QACI,IAAI,KAAM,WAAN,IAAqB,IAAK,WAAL,KAakB,KAAM,WAaxB,gBAAoC,CAA7D,C;UACI,OAAO,I;;UAEP,MAAM,gCAAYB,2EAAzB,C;WAEd,IAAA,KAAM,aAAN,C;QAAsB,OAAO,K;MAI7B,KAx0C,eAAS,QAAT,GAAqB,CAwb/D,OAA0B,KAxgbB,WAAS,QAAT,GAAqB,CAwb/D,E;QACI,aAAa,IAAK,QAAL,KAAa,KAAM,QAAnB,C;QAET,uB;UACI,iCAA0B,MAA1B,C;;UAEA,kCAA2B,MAA3B,C;aAGZ,IAAA,IAAK,eAAL,C;QACI,mCAAqB,IAAK,QAA1B,EAAiC,KAAM,QAAvC,C;;QAEA,mCAAqB,KAAM,QAA3B,EAAkC,IAAK,QAAvC,C;MABr,W;K;gDAiBJ,kC;MAGW,Q;MAFP,kBAakB,cAAc,UAAc,C;MACIB,mBAAmB,eAAa,WAAb,C;MACZ,IAAI,8EAAc,mBAAT,C,CAAJ,C;QACH,yBAAYB,oBAAa,cAAc,WAAd,CAAb,C;QACzB,uBAAGB,cAAc,YAAc,MAA8B,kBAA9B,CAAhB,C;;QAEA,wBAA8B,WAAb,YAAa,yBAAsB,UAAtB,CAA9B,C;;MAJJ,W;K;sCAQJ,iB;MAMuD,wBAAS,KAAD,aAAR,C;K;uCAEvD,iB;MAQe,UAUJ,M;MAXP,IAAI,iBAAJ,C;QAEQ,cAAS,CAAT,C;UAAc,MAAM,gCAAYB,mEAAzB,C;aCpB,YAAQ,CAAR,C;UAAa,W;;UACL,OAC,IAAD,a;QAHZ,W;;MAMJ,IAAI,UAAS,CAAb,C;QAAGB,OAAO,qC;MAEvB,YAAy,Y;MACZ,aAAa,mCAAQ,KAAR,E;MACN,IAAI,kBAAJ,C;QACH,IAAI,yEAAJ,C;UAEI,yBAAGB,MAAhB,C;;UAEA,IAAI,sCAAS,KAAAT,IAAkB,KAAIB,CAAJ,C;YACI,mCAA0B,MAA1B,C;;YAEA,aAAa,cAAc,KAAd,C;YACb,eAAe,eAAQ,cAAc,MAAd,CAAR,C;YACf,mBAAmB,oCAAS,KAAT,E;YACnB,kBAakB,iBA Ae,cAAc,sCAAW,KAAX,EAAd,CAAf,C;YACIB,IAAI,4CAAE,KAAf,IAAwB,MAAxB,KAakC,gBAAGB,YAAhB,gBAAGC,CAATe,C;cACI,0BAA6B,WAAZ,WAAY,EAAS,8BAAa,UAAb,CAAT,CAA7B,C;;cAEA,SAAI,YAAM,WAAN,KAAM,CAAN,EAAMB,WAAN,KAAM,CAANB,IAA0B,CAA9B,GAAiC,yCAAjC,GAA+C,qD;;;QAK3D,IAAI,sCAAS,KAAT,IAAkB,KAAIB,CAAJ,C;UACI,0BAAwB,WAAP,MAAO,EAAS,8BAAa,UAAb,CAAT,CAAxB,C;;UAEA,SAAI,YAAM,WAAN,KAAM,CAAN,EAAMB,WAAN,KAAM,CAANB,IAA0B,CAA9B,GAAiC,yCAAjC,GAA+C,qD;;;MAVbD,a;K;uCA4BJ,iB;MASI,eAAqB,WAAN,KAAM,C;MACrB,IAAa,QAAT,KAAuB,KAA3B,C;QACI,OAAO,mBAAM,QAAN,C;;MAGX,WAAW,kB;MACX,aAAa,sBAAS,IAAT,IAAiB,K;MAC9B,OAAc,aAAP,MAAO,EAAW,IAAX,C;K;qCAGIB,iB;MAQe,Q;MADX,IAAI,UAAS,CAAb,C;QAEQ,sB;UAAgB,gD;aAchB,sB;UAAgB,4D;;UACR,MAAM,gCAAYB,4DAAZB,C;QAHIB,W;;MAMJ,IAAI,kBAAJ,C;QACI,OAAO,gBAAGB,qCAAQ,KAAR,EAAhB,C;;QAEp,IAAI,iBAAJ,C;UACI,OAAO,mBAAa,WAAN,KAAM,CAAb,C;QAEX,aAAa,qCAAQ,KAAR,E;QAEb,IAAI,kEAAgC,mBAAhC,CAAJ,C;UACI,UAAU,cAAc,sBAAS,oCAAS,KAAT,EAAT,CAAd,0BAA0C,KAA1C,E;UACV,OAAO,gBAAGB,cAAc,MAAd,MAAwB,GAAxB,CAAhB,C;;QAEX,OAAO,iBAAiB,MAAjB,C;;K;qCAIf,iB;MAOI,eAAqB,WAAN,KAAM,C;MACrB,IAAa,QAAT,KAAuB,KAAvB,IAAgC,aAAY,CAAhD,C;QACI,OAAO,iBAAI,QA AJ,C;;MAGX,WAAW,kB;MACX,aAAa,sBAAS,IAAT,IAAiB,K;MAC9B,OAAc,aAAP,MAAO,EAAW,IAAX,C;K;oCAGIB,iB;MAEI,kBAakB,SAAM,IAAK,cAAX,EAawB,KAAM,cAA9B,C;MACIB,OAAO,IAAK,kBAAS,WAAT,CAAL,GAA6B,KAAM,kBAAS,WAAT,C;K;oCAG9C,Y;MACmC,oCAAW,C;K;oCAE9C,Y;MACmC,oCAAW,C;K;oCAE9C,Y;MACmC,+BAAY,yCAAS,WAARb,KAAiC,wBAAY,qDAAa,WAAzB,C;K;kCAEpE,Y;MACiC,QAAC,iB;K;yFAGC,Y;MAAQ,OAAI,iBAAJ,GAAMB,IAAD,aAaIB,GAA6B,I;K;yCAExE,iB;MACI,kBAakB,IAAK,WAAL,KAakB,KAAM,WAaxB,C;MACIB,IAAI,yBAAc,CAAd,IAAmB,CAAA,WAAY,QAAZ,GAAwB,CAAxB,MAA6B,CAApD,C;QACI,OAAO,IAAK,WAAS,iBAAU,KAAM,WAAhB,C;MAEzB,QA AQ,CArmBsC,eAAS,QAAT,GAAqB,CAqmB3D,KAAyB,KArmBa,WAAS,QAAT,GAAqB,CAqmB3D,K;MACR,

OAAW,iBAAJ,GAaKB,CAAC,CAAD,IAAIB,GAA0B,C;K;uHAMrC,kB;MAeI,OAAO,OAAO,gBAAP,EAAoB,mBAAPB,EAAoC,qBAAPC,EAAAsD,qBAAtD,EAAwE,yBAAxE,C;K;uHAGX,kB;MAcI,OAAO,OAAO,iBAAP,EA AqB,qBAArB,EAAuC,qBAAvC,EAAyD,yBAAzD,C;K;uHAGX,kB;MAaI,OAAO,OAAO,mBAAP,EAAuB,qBAA vB,EAAyC,yBAAzC,C;K;uHAGX,kB;MAYI,OAAO,OAAO,mBAAP,EAAuB,yBAAvB,C;K;0FAKP,Y;MAAQ,O AAI,iBAAJ,GAaKB,CAAIB,GAA0B,6CAAe,EAaf,EAAMB,Q;K;4FAIrD,Y;MAAQ,OAAI,iBAAJ,GAaKB,CAAI B,GAA0B,+CAAiB,EAajB,EAaqB,Q;K;4FAIvD,Y;MAAQ,OAAI,iBAAJ,GAaKB,CAAIB,GAA0B,+CAAiB,EA AjB,EAaqB,Q;K;gGAIvD,Y;MACI,sB;QADI,OACY,C;WACHb,wB;QAFI,OAeY,cAAc,wCAAQ,IAAR,EAAd,C AA6B,Q;QAFzC,OAGK,wCAAQ,UAAR,EAAuB,Q;K;0CAMxC,gB;MAQiB,UAAN,M;MAAM,sB;MACT,iBA AA,yCAAS,WAAT,E;QAA4B,SAAP,wCAAO,kB;WAC5B,iBAAA,qDAAa,WAAb,E;QAAgC,SAAP,wCAAO,kB ;;QAG5B,6BAAoB,YAAM,WAA1B,EAAcS,kBAAtC,EAAMd,IAANd,C;MALR,a;K;wCAUI,gB;MAUiB,UA AN,M;MAAM,sB;MACT,iBAAA,yCAAS,WAAT,E;;WACA,iBAAA,qDAAa,WAAb,E;;;QACQ,+BAAoB,YAAPB, EAA2B,kBAA3B,EAAwC,IAAxC,C;MAHZ,a;K;uCAOJ,gB;MAUI,OAAa,WAAb,oBAAO,IAAP,CAAa,4BAAyD ,Q;K;kFAKhD,Y;MAAQ,6D;K;mFAKP,Y;MAAQ,8D;K;qFAKN,Y;MAAQ,gE;K;qFAKR,Y;MAAQ,gE;K;0FAK H,Y;MAAQ,qE;K;0FAKR,Y;MAAQ,qE;K;yFAKT,Y;MAAQ,oE;K;uFASrC,Y;MAAQ,2D;K;wFAQR,Y;MAAQ,4 D;K;0FAQR,Y;MAAQ,8D;K;0FAQR,Y;MAAQ,8D;K;+FAQR,Y;MACI,OAAW,uBAAgB,eAApB,GAAGC,YAAh C,GAA2C,4D;K;+FAAtD,Y;MAAQ,mE;K;8FAYR,Y;MAEW,Q;MADP,YAA Y,Y;MAER,uB;QAae,Y;WAcF,8C;; WACA,+C;;;QACQ,qBAAc,KAAAd,C;MAJZ,W;K;2CAUR,Y;MASuC,8B;K;4CAEvC,Y;MASwC,+B;K;kCAExC, Y;MAuBwC,Q;MAAA,sB;MACpC,qB;QAD8B,OACxB,I;WACN,iBAAA,yCAAS,WAAT,E;QAF8B,OAET,U;W ACrB,iBAAA,qDAAa,WAAb,E;QAH8B,OAGL,W;;QAErB,iBAAiB,iB;Q2HpiBF,gBAAhB,sB;Q3HsiBK,e;UAA gB,yBAAO,EAAP,C;QACF,YAAAd,kB;QA9RD,WAAO,iB;QAAP,YAAoB,oB;QAAPB,cAAoC,sB;QAAPC,cAAs D,sB;QAAtD,kBAAwE,0B;QAsS/D,0B;QAPJ,cAAc,iB;QACd,eAAe,UAAS,C;QACxB,iBAAiB,YAAW,C;QAC5 B,iBAAiB,YAAW,CAAX,IAAgB,gBAAe,C;QAChD,iBAAiB,C;QACjB,IAAI,OAAJ,C;UACI,yBAAO,IAAP,CA Aa,gBAAO,GAAP,C;UACb,+B;;QAEJ,IAAI,aAAa,YAA Y,cAAc,UAA1B,CAAb,CAAJ,C;UACI,IAAI,6DAAe,C AAnB,C;YAAsB,yBAAO,EAAP,C;UACtB,yBAAO,KAAP,CAAc,gBAAO,GAAP,C;;QAEIB,IAAI,eAAe,eAAe,Y AAY,OAA3B,CAAf,CAAJ,C;UACI,IAAI,6DAAe,CAnB,C;YAAsB,yBAAO,EAAP,C;UACtB,yBAAO,OAAP,C AAgB,gBAAO,GAAP,C;;QAEpB,IAAI,UAAJ,C;UACI,IAAI,6DAAe,CAnB,C;YAAsB,yBAAO,EAAP,C;UAEI B,gBAAW,CAAX,IAAgB,OAAhB,IAA2B,QAA3B,IAAuC,UAAvC,C;YACI,mCAAiB,OAAjB,EAA0B,WAA1B, EAAuC,CAAvC,EAA0C,GAA1C,EAA2D,KAA3D,C;eACJ,mBAAe,OAAf,C;YACI,mCAAiB,cAAc,OAAd,IAAj B,EAA0C,cAAc,OAAd,IAA1C,EAAMe,CAnE,EAASe,IAAtE,EAAwF,KAAxF,C;eACJ,mBAAe,IAAf,C;YACI, mCAAiB,cAAc,IAAd,IAAjB,EAAc,cAAc,IAAd,IAAtC,EAA2D,CAA3D,EAA8D,IAA9D,EAAGF,KAAhF,C;;Y AEA,yBAAO,WAAP,CAAoB,gBAAO,IAAP,C;;QAGhC,IAAI,cAAc,aAAa,CAA/B,C;UAAkC,yBAAO,CAAP,EA AU,EAAV,CAAE,gBAAO,EAAP,C;QAvC/B,OQn2B3B,SmHoUqC,W;;K;4C3H4k5C,yE;MACI,yBAAO,KAAP, C;MACA,IAAI,eAAc,CAAIB,C;QACI,yBAAO,EAAP,C;QACA,iBAAuC,WAATB,UAAW,WAAW,EAAS,cAAT, EAAyB,EAAzB,C;QACR,sB;;UuBt0BzB,Q;UAAA,OAAQ,WAAR,evBs0Bc,UuBt0Bd,CAAQ,CAAR,W;UAAAd,O AAc,cAAAd,C;YAAc,uB;YACV,IvBq0BiD,UuBr0BnC,YvBq0BU,UuBr0BV,YAAK,KAAL,EvBq0BmC,MAAM,E uBr0BvD,C;cACI,qBAAO,K;cAAP,uB;;UAGR,qBAAO,E;;QvBi0BC,oBAAoB,qBAAuC,CAAvC,I;QAEhB,KA AC,SAAD,IAAc,gBAAgB,CAA9B,C;UAAmC,8BAAY,UAAZ,EAAwB,CAAxB,EAA2B,aAA3B,C;;UAC3B,8BA AY,UAAZ,EAAwB,CAAxB,EAA2B,CAAC,CAAC,gBAAgB,CAAhB,IAAD,IAAsB,CAAtB,IAAD,IAA4B,CAA5 B,IAA3B,C;;MAGhB,yBAAO,IAAP,C;K;0CAGJ,0B;MAGBwC,wB;QAAA,WAAgB,C;MK99BxD,IAAI,EL+9BQ ,YAA Y,CK/9BpB,CAAJ,C;QACI,cL89ByB,oD;QK79BzB,MAAM,gCAAYB,OAAQ,WAAjC,C;;ML89BN,aAAa,s BAAS,IAAT,C;MACb,IAAW,WAAP,MAAO,CAAX,C;QAAYB,OAAO,MAAO,W;MACvC,OAAO,sBAAsB,MA AtB,EAAuC,eAAT,QAAS,EAAa,EAAb,CAAvC,IAAGe,UAAAL,IAAK,C;K;qCAI3E,Y;M2HlnBuB,gBAAhB,sB; M3HgoBH,IAAI,iBAAJ,C;QAAkB,yBAAO,EAAP,C;MACIB,yBAAO,IAAP,C;MAC4B,YAAAd,kB;MAxWP,YAA O,kB;MAAP,cAAqB,sB;MAArB,cAAuC,sB;MAAvC,kBAAyD,0B;MAyW5D,cACY,K;MACZ,IAAI,iBAAJ,C;Q AEI,wB;;MAEJ,eAAe,oB;MACf,iBAAiB,YAAW,CAAX,IAAgB,gBAAe,C;MACHD,iBAAiB,YAAW,CAAX,KA AiB,cAAc,QAA/B,C;MACjB,IAAI,QAAJ,C;QACI,yBAAO,OAAP,CAAc,gBAAO,EAAP,C;;MAEIB,IAAI,UAAJ, C;QACI,yBAAO,OAAP,CAAgB,gBAAO,EAAP,C;;MAEpB,IAAI,eAAe,CAAC,QAAD,IAAa,CAAC,UAA7B,CA AJ,C;QACI,mCAAiB,OAAjB,EAA0B,WAA1B,EAAuC,CAAvC,EAA0C,GAA1C,EAA2D,IAA3D,C;;MApBuB,O

Qn8B5B,SmHoUqC,W;K;;;;;kC3H5YhD,Y;MAAA,c;MAuBiD,2D;MAvBjD,a;K;gCAAA,iB;MAAA,2IAuBiD,gD  
AvBjD,G;K;IAyiCA,qC;MAIW,Q;MAAA,IAAI,6DAAJ,C;QACH,uBAAGB,4BAAiC,oBAAL,SAAK,CAAjC,EA  
A2C,IAA3C,yCAAhB,C;;QAES,oBAAT,8BAAS,EAAW,IAAX,C;MAHb,W;K;IAMJ,uC;MAII,kBAaKB,4BAA4  
B,SAA5B,0CAAiE,IAAJE,C;MACiB,IAAa,WAAD,aAAR,yDAAsB,WAAtB,CAAJ,C;QACI,OAAO,gBAAGB,4B  
AA4B,SAA5B,EAaKc,IAAIC,yCAAhB,C;;QAEP,aAAa,sBAAoB,SAApB,EAA0B,IAA1B,0C;QACb,OAAO,iBA  
AwB,WAAP,MAAO,yBAAsB,UAAAtB,CAAxB,C;;K;IAIf,uC;MAAw,Q;MAHP,gBAAGB,oBAAoB,SAApB,EAA0  
B,IAA1B,yC;MKljChB,IAAI,CLmjCI,CAAW,QAAV,SAAU,CKnjCnB,C;QACI,cLkjC0B,+B;QKjjC1B,MAAM,g  
CAAyB,OAAQ,WAAjC,C;;MLkjCV,YAAsB,YAAV,SAAU,C;MACf,IAAI,sEAAqB,SAArB,CAAJ,C;QACH,uB  
AAgB,KAAhB,C;;QAEA,aAAwE,YAA3D,oBAAoB,SAApB,EAA0B,IAA1B,0CAA2D,C;QACxE,kCAA2B,MAA  
3B,C;;MAJJ,W;K;IAGBuB,oC;MAAQ,oE;K;IAOP,sC;MAAQ,sE;K;IAWN,sC;MAAQ,sE;K;IAQV,qC;MAAQ,qE;  
K;IAOP,uC;MAAQ,uE;K;IAWN,uC;MAAQ,uE;K;IAQX,qC;MAAQ,qE;K;IAOP,uC;MAAQ,uE;K;IAWN,uC;MA  
AQ,uE;K;IAQhB,gC;MAAQ,gE;K;IAOP,kC;MAAQ,kE;K;IAWN,kC;MAAQ,kE;K;IAQX,gC;MAAQ,gE;K;IAOP,  
kC;MAAQ,kE;K;IAWN,kC;MAAQ,kE;K;IAQb,8B;MAAQ,8D;K;IAOP,gC;MAAQ,gE;K;IAWN,gC;MAAQ,gE;K  
;IAQZ,6B;MAAQ,6D;K;IAOP,+B;MAAQ,+D;K;IAWN,+B;MAAQ,+D;K;yEAG/B,+B;MAIqE,8BAAW,SAAX,C;  
K;2EAERe,+B;MAUwE,8BAAW,SAAX,C;K;IAIx,yC;MACI,aAAa,KAAM,O;MACnB,IAAI,WAAU,CAAd,C;Q  
AAiB,MAAM,gCAAyB,qBAAzB,C;MACvB,YAAy,C;MACZ,aAAa,gCAAS,K;MACtB,qBAaQb,U;MACrB,QA  
AM,iBAAM,KAAN,CAAN,C;aACI,E;aAAa,E;UAAy,qB;UAAZ,K;;MAEJ,cAAc,QAAQ,C;MACtB,iBAAiB,WA  
AiB,aAN,KAAM,EAAW,EAAX,C;MAE9B,cAAU,KAAY,C;QACI,MAAM,gCAAyB,eAAzB,C;WACV,qBAA  
M,KAAN,MAAGB,EAAhB,C;QACI,IAAI,mCAAW,MAAf,C;UAAuB,MAAM,+B;QAC7B,sBAAsB,K;QACtB,sB  
AAsB,K;QACtB,eAA8B,I;QAC9B,OAAO,QAAQ,MAAf,C;UACI,IAAI,iBAAM,KAAN,MAAGB,EAAPB,C;YAC  
I,IAAI,mBAAmB,mCAAW,MAAIC,C;CAA0C,MAAM,+B;YACHd,kBAaKB,I;YACIB,Q;;UAekB,iBAaE,K;UA+  
EjD,QAHgC,U;UAIhC,Y;YAAO,eAhFqB,KAGfjB,O;YAAJ,S;cAAc,SAAU,YAhFH,KAGFG,YAAK,CAAL,E;cA  
AV,OAhFqC,CAAM,kBAAK,EAAL,CAAN,qCAAKB,2C;;;YAgFnC,a;;UAhF7B,gBAAGB,KkBzCgE,WIBqpCl  
F,UkBrpCkF,ElB0pCrF,CkB1pCqF,C;UIB0kChF,IAAI,SwBhiCgC,YAAU,CxBgiC9C,C;YAAyB,MAAM,+B;UA  
C/B,gBAAS,SAAU,OAAAnB,I;UACqB,cAAU,K;UuBnsCpC,U;UAAA,IAAI,WAAS,CAAT,IAAc,WAAS,iBvBms  
CP,KuBnsCO,CAA3B,C;YAAA,SvBmsCoB,KuBnsCkB,YAAI,OAAJ,C;;YvBmsCO,MAAM,gCAAyB,qCAAzB,  
C;;UAA9C,qB;UACA,qB;UACA,WAAW,sBAAsB,QAAtB,EAAGC,eAAhC,C;UACX,IAAI,YAAy,IAAZ,IAAoB,  
yBAAY,IAAZ,MAAxB,C;YAA0C,MAAM,gCAAyB,yCAAzB,C;UACHd,WAAW,I;UACX,eAAyB,WAAV,SAA  
U,EAAQ,EAAR,C;UACzB,IAAI,+CAAGC,WAAW,CAA/C,C;YACI,YAAy,SkBnlCgE,WIBmlC5C,CkBnlC4C,El  
BmlCzC,QkBnlCyC,C;YIBolC5E,4BAA2C,aAAjC,0BAA0B,KAA1B,CAAiC,EAAW,IAAX,CAA3C,C;YACA,4B  
AAmD,aAAx,SAA9B,SkBxlCmD,WIBwlC/B,QkBxlC+B,CIBwlCrB,CAAW,EAAW,IAAX,CAANd,C;;YAEA,4B  
AA+C,aAArC,0BAA0B,SAA1B,CAAqC,EAAW,IAAX,CAA/C,C;;aAIZ,c;QACI,MAAM,+B;;QACV,IAAM,cAA  
N,KAAM,EAAC,KAAd,EAAqB,cAArB,EAAqC,CAArC,ES1yCH,MAAO,KT0yCmD,SAAS,KAAT,IS1yCnD,ET  
0yCmE,cAAe,OS1yClF,CT0yCJ,EAA4G,IAA5G,CAAN,C;UACI,SAAS,gCAAS,S;;UAIIB,iBAA8B,I;UAC9B,iB  
AAiB,K;UACjB,kBAaKB,CAAC,O;UACnB,IAAI,WAAW,iBAAM,KAAN,MAAGB,EAA3B,IAAwC,QAAN,KA  
AM,CAAN,KAAGB,EAAtD,C;YACI,cAAc,I;YACd,IAAI,oCAAW,uBAAX,EAAW,MAAX,CAAJ,C;cAAyB,MA  
AM,gCAAyB,eAAzB,C;;UAEnC,OAAO,QAAQ,MAAf,C;YACI,IAAI,cAAc,WAAIB,C;cA8CZ,UA7CwC,K;cA8  
CxC,Y;gBAAO,mBA9CiB,KA8Cb,O;gBAAJ,W;kBAAc,SA9C4B,UA8CIB,YA9CP,KA8CO,YAAK,GAAL,EA9C  
kB,MAAM,E;;;gBA8Cd,iB;;cA9CzB,QA+CT,G;;YA7CK,aAAa,I;YACS,mBAaE,K;YA0CjD,UAHgC,Y;YAIhC,  
Y;cAAO,mBA3CqB,KA2CjB,O;cAAJ,W;gBAAc,WAAU,YA3CH,KA2CG,YAAK,GAAL,E;gBAAV,SA3CqC,C  
AAM,kBAAK,EAAL,CAAN,uCAAKB,oBAAM,E;;;cA2CzC,iB;;YA3C7B,kBAAGB,KkB9mCgE,WIBqpClF,YkB  
rpCkF,ElB0pCrF,GkB1pCqF,C;YIB+mChF,IAAI,WwBrkCgC,YAAU,CxBqkC9C,C;cAAyB,MAAM,+B;YAC/B,g  
BAAS,WAAU,OAAAnB,I;YACqB,mBAaE,K;YAUChD,UAHgC,Y;YAIhC,Y;cAAO,mBAxCoB,KAwChB,O;cAAJ  
,W;gBAAc,WAAU,YAxCJ,KAwCl,YAAK,GAAL,E;gBAAV,SAxCoC,CAAM,kBAAK,GAAL,CAAN,mC;;;cAw  
ChB,iB;;YAx7B,eAAe,KkBjnCiE,WIBqpClF,YkBrpCkF,ElB0pCrF,GkB1pCqF,C;YIBknChF,gBAAS,QAAS,OA  
AlB,I;YACA,aAAW,wBAAwB,QAAXB,C;YACX,IAAI,cAAy,IAAZ,IAAoB,2BAAY,MAAZ,MAAxB,C;cAA0C,  
MAAM,gCAAyB,yCAAzB,C;YACHd,aAAW,M;YACX,iBAAYB,WAAV,WAAU,EAAQ,EAAR,C;YACzB,IAAI,  
aAAW,CAAF,C;cACI,cAAy,WkBxnCgE,WIBwnC5C,CkBxnC4C,ElBwnCzC,UkBxnCyC,C;clBynC5E,4BAAYB,

aAAT,OAAN,OAAM,CAAS,EA AW,MAAX,CAAzB,C;cACA,4BAAmD,aAAX,SAA9B,WkB7nCmD,WIB6nC/B,  
Uk b7nC+B,CIB6nCrB,CAAW,EA AW,MAAX,CAAnD,C;cACA,IAAI,QAAQ,MAAZ,C;gBAAoB,MAAM,gCAA  
yB,mCAAzB,C;;cAE1B,4BAA6B,aAAT,OAAV,WAAU,CAAS,EA AW,MAAX,CAA7B,C;;;;MAKhB,OA AW,U  
AAJ,GAAiB,MAAD,aAAhB,GAA6B,M;K;IAIx C,0C;MACI,aAAa,KAAM,O;MACnB,iBAAiB,C;MACjB,IAAI,S  
AAS,CAAT,IAAc,YAAY,IAAZ,mBAAM,CAAN,EAAiB,C;QAAoC,+B;;MACHC,YAAC,SAAS,UAAT,IAAD,IA  
AwB,E;MAAx B,S;QAA4D,gBAA7B,yBAAkB,iBAAN,KAAM,CAAiB,C;QAA6B,c;;UWmThD,U;UADhB,IAAI,  
wCAAsB,mBAA1B,C;YAAqC,aAAO,I;YAAP,e;;UACrB,6B;UAAhB,OAAgB,gBAAhB,C;YAAgB,2B;YAAM,IA  
AI,CXnT4C,CAAa,kBAAK,EAAL,CAAb,oCWmTjC,OXnTiC,EWmThD,C;cAAyB,aAAO,K;cAAP,e;;;UAC/C,a  
AAO,I;;;QXpTyD,iB;;MAAhE,S;QAEI,OA AW,iBAAM,CAAN,MAAY,EAAhB,sD;;MAGX,OAAiB,WAAN,KA  
AM,EA AW,GAAX,CAAV,GAAYC,OAAR,QAAN,KAAM,EA AK,CAAL,CAAQ,CAAzC,GAA6D,OAAN,KAA  
M,C;K;IAKxE,0D;MAII,QAHgC,U;MAIhC,OAAO,IAAI,gBAAJ,IAJqC,SAIvB,CAAU,iCAAk,CAAL,EA AV,CA  
ArB,C;QAAyC,a;;MAJzC,OkBrpC4F,oBlBqpCIF,UkBrpCkF,ElB0pCrF,CkB1pCqF,C;K;IlBupChG,qD;MACI,QA  
AQ,U;MACR,OAAO,IAAI,gBAAJ,IAAc,UAAU,iCAAk,CAAL,EA AV,CAArB,C;QAAyC,a;;MACzC,OAAO,C;  
K;;;;IAmBX,8B;MAA+C,qCAAQ,OAAR,E;K;IAC/C,+B;MAAgD,2CAAS,OAAT,E;K;IAEHd,sC;MAAiD,oBAA  
S,sBAAGB,CAAhB,CAAT,C;K;IACjD,wC;MAAmD,oBAAU,uBAAiB,CAAjB,CAAD,yBAAuB,CAAvB,EAAT,  
C;K;IACnD,oD;MAAOE,oBAAU,sBAAGB,CAAhB,CAAD,yBAAsB,iBAAtB,EAAT,C;K;IACpE,0C;MACI,IAAI,  
sEAAqB,SAArB,CAAJ,C;QAAA,OACI,gBAAgB,KAAhB,C;;QADJ,OAGI,iBAAiB,cAAc,KAA d,CAAjB,C;;K;IA  
GR,4C;MACI,IAAI,kEAAgC,mBAAhC,CAAJ,C;QAAA,OACI,gBAAgB,cAAc,MAAd,CAAhB,C;;QADJ,OAGI,i  
BAAwB,WAAP,MAAO,yBAAsB,UAAtB,CAAxB,C;;K;I0M73CR,8B;MAEGD,QAAM,SAAN,M;aAC5C,a;UAD  
4C,OACHB,I;aAC5B,c;UAF4C,OAef,I;aAC7B,c;UAH4C,OAGf,I;aAC7B,S;UAJ4C,OAIpB,G;aACxB,S;UAL4C,  
OAKpB,G;aACxB,O;UAN4C,OAMtB,G;aACtB,M;UAP4C,OAovB,G;;UrMuEwB,MAAM,6BAA8B,CqMtEnE,m  
BAAGB,SrMsEmD,YAA9B,C;;K;IqMnEvD,4C;MACwE,QAAM,SAAN,C;aACpE,I;UADoE,6C;aAEpE,I;UAFoE,  
8C;aAGpE,I;UAHoE,8C;aAlpE,G;UAJoE,yC;aAKpE,G;UALoE,yC;aAMpE,G;UANoE,uC;aAOpE,G;UAPoE,sC;;  
UAQ5D,MAAM,gCAAyB,uCAAoC,SAA7D,C;;K;IAGIb,yD;MAGQ,KAAC,eAAD,C;QAEQ,IADE,OACF,Q;UA  
HZ,sC;;UAIoB,MAAM,gCAAyB,4EAAqD,OAARd,CAAzB,C;;QAIIB,QAAM,OAAN,C;eACI,E;YATZ,uC;eAUY  
,E;YAVZ,yC;eAWY,E;YAXZ,yC;;YAYoB,MAAM,gCAAyB,yDAAKC,OAAIC,CAAzB,C;;K;IC5F9B,4B;K;;;M  
CqDI,kC;;IAICA,gC;MAAA,oC;K;6CAUI,Y;MAAwC,OAAA,iCAAoB,U;K;8CAC5D,Y;MAAkC,OAAA,iCAAo  
B,W;K;IACrB,qD;MAAqB,8B;K;8DACID,Y;MAAsC,OAAA,iCAAoB,qBAAy,IAAZ,C;K;+DAC1D,oB;MAAuD,  
OAAA,iCAAoB,uBAAc,IAAd,EAAoB,QAAPB,C;K;gEAC3E,oB;MAAwD,OAAA,iCAAoB,uBAAc,IAAd,EAAq  
B,QAAD,aAAPB,C;K;gEAC5E,Y;MAAuC,QAAC,iBAAa,a;K;mEACrD,Y;MAA0C,OAAA,iBAAa,a;K;;;;4DAjB  
3D,Y;MAAA,OAYsD,gEAZtD,M;K;4DAAA,Y;MAAA,c;MAYsD,gE;MAZtD,a;K;0DAAA,iB;MAAA,2IAYsD,0  
DAZtD,G;K;;;IABJ,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;IAkCA,gC;MAAA,oC;K;;;IAAA,4C;MAAA,2C;QA  
AA,0B;;MAAA,oC;K;;;qCA2BA,oB;MAW8D,4BAAiB,IAAjB,EA AuB,QA AvB,C;K;sCAE9D,oB;MAW+D,wBA  
AM,QAAD,aAAL,C;K;sCAG/D,Y;MAMqC,QAAC,iBAAa,a;K;yCAEnD,Y;MAMwC,OAAA,iBAAa,a;K;;4EAIz  
D,yB;MAAA,4C;MAAA,mC;QAQuE,MAAM,WAAM,0BAAN,C;O;KAR7E,C;mFAUA,yB;MAAA,4C;MAAA,m  
C;QAQsE,MAAM,WAAM,0BAAN,C;O;KAR5E,C;IAY8B,4C;MAAC,gB;MAAoB,4B;K;4CAC/C,Y;MAAsC,OA  
AA,SAAK,aAAL,cAAoB,eAAPB,C;K;6CAEtC,oB;MAAkD,4BAAiB,SAAjB,EA AuB,4BAAa,QAAb,CAA vB,C;  
K;;ICzIV,sC;MAAC,gB;K;IAOf,4E;MAAC,4B;MAA6B,8B;MAAgD,sB;K;+DACpG,Y;MAAsC,OAAgC,AA/B,i  
BAAW,OAA X,UAAoB,gBAApB,CAA+B,EA AW,iBAAW,KAAtB,CAAhC,cAA8D,AA9D,C;K;gEACtC,oB;MA  
AkD,+CAAa,gBAAb,EA AwB,iBAAxB,EAAoC,0BAAS,QAAT,CAAPC,C;K;;+CAGtD,Y;MAAmC,+CAAa,WA  
Ab,EAAqB,IAArB,EAA2B,gCAAS,KAAPC,C;K;;IAUO,wC;MAAC,gB;K;IAOf,gF;MAAC,4B;MAA+B,8B;MAA  
kD,sB;K;mEAC1G,Y;MAAsC,OAAgC,AA/B,iBAAW,OAA X,GAAoB,gBAAW,EA AW,iBAAW,KAAtB,CAAh  
C,cAA8D,AA9D,C;K;oEACtC,oB;MAAkD,mDAAe,gBAAf,EAA0B,iBAA1B,EAAsC,0BAAS,QAAT,CAATC,C;  
K;;iDAGtD,Y;MAAmC,mDAAe,WAAf,EAAuB,IAAvB,EAA6B,gCAAS,KAAtC,C;K;;IAGvC,0B;MAGB8B,yE;  
MAC1B,mB;K;oCAEA,Y;MAA4B,qB;K;iDAE5B,oB;MAWc,Q;MADV,gBAAgB,QAAS,gBAAO,SAAP,C;MACf  
,IAAI,gDAA+B,4CAAnC,C;QAEN,iBAAiB,mBAAU,SAAV,C;QACjB,IAAI,mBAAy,SAAZ,gBAAyB,CAAzB,I  
AA8B,mBAAy,UA AZ,eAAyB,CAA3D,C;UAA8D,gBAAS,QAAT,C;QAC9D,iB;;QAEA,YAAY,QAAS,kBAAS,  
SAAT,C;QAErB,mBAAiB,4BAAU,K;QAC3B,IAAI,sDAA+B,kDAAnC,C;UAAgE,gBAAS,QAAT,C;QACrD,8B

AAX,YAAW,C;;MAVf,qB;K;0CAcJ,oB;MACI,MAAM,6BAAsB,iDAA+C,cAA/C,qCAA0E,QAA1E,MAAtB,C;K  
;;IC9Fd,yC;MACI,iBAAiB,QAAS,mB;MAC1B,IAAI,OAAC,oCAAS,CAAT,EAAD,kCAAJ,C;QACI,OAAO,wBA  
AwB,MAAxB,EAAgC,QAAhC,EAA0C,UAA1C,C;;MAEX,IAAI,OAAC,wCAAa,CAAAb,EAAD,kCAAJ,C;QACI,  
OAAO,sBAAsB,MAAtB,EAA8B,QAA9B,C;;MAGX,aAAa,WAAS,UAAT,C;MACb,IAAM,WAAW,MAAX,CAA  
D,KAAyB,eAAe,MAAf,CAAzB,CAAD,cAAoD,CAAxD,C;QACI,OAAW,oBAAS,CAAAb,sD;;MAEX,OAAO,M;K  
;IAGX,+D;MACI,IAAI,QAAS,aAAT,IAA0B,WAAW,UAAx,eAAwB,CAAtD,C;QAA0D,MAAM,gCAAyB,uCA  
AzB,C;MACHe,OAAO,M;K;IAGX,iD;MACI,WAAW,qBAAW,CAAX,C;MACX,IAAI,OAAC,IAAK,mBAAL,8B  
AA0B,CAA1B,EAAD,kCAAJ,C;QAEI,OAA8D,uBAAtD,oBAAS,QAAS,yDAAoC,C;;QAE9D,OAAO,cAAc,cAA  
c,MAAd,EAAsB,IAAtB,CAAd,EAA2C,IAA3C,C;;K;IAIf,2C;MACI,IAAI,OAAC,sCAAW,CAAX,EAAD,kCAAJ,  
C;QACI,OAAkB,aAAT,QAAS,kCAAX,a;;MAEX,aAAa,iBAAU,QAAY,C;MACb,IAAK,WAAW,OAAx,CAAD,  
KAA0B,WAAW,QAAX,CAAqB,MAA/C,eAAuD,CAA3D,C;QACI,eAAe,gCAAU,OAAV,YAA4B,iCAAW,OAA  
X,EAA5B,C;QACf,eAAe,mCAAU,OAAV,YAA4B,oCAAW,OAAx,EAA5B,C;QACf,O9M6D4C,a8M7DrC,Q9M  
6DqC,4B8M7DrC,a9MuBoC,a8MvBZ,Q9MuBY,2B8MvBpC,C;;MAEX,O9MqB+C,a8MrBxC,M9MqBwC,2B;K;q  
F+MjEnD,yB;MAAA,yC;MAAA,wB;QA4CI,WAAW,8B;QAjC6B,KakCxC,E;QAICA,OAmCO,IAAK,a;O;KA9C  
hB,C;uFAeA,4B;MAYI,WAAW,mB;MACX,O;MACA,OAAO,IAAK,a;K;uFAGhB,4B;MAYI,WAAW,mB;MAC  
X,O;MACA,OAAO,IAAK,a;K;IAYe,qC;MAAC,kB;MAAc,wB;K;;sCAR9C,Y;MAQgC,iB;K;sCARhC,Y;MAQ8C  
,oB;K;wCAR9C,2B;MAAA,sBAQgC,qCARhC,EAQ8C,8CAR9C,C;K;oCAAA,Y;MAAA,OAQgC,iDARhC,IAQ8  
C,8CAR9C,O;K;oCAAA,Y;MAAA,c;MAQgC,sD;MAR9C,a;K;kCAAA,iB;MAAA,4IAQgC,sCARhC,I  
AQ8C,4CAR9C,I;K;iGAUA,yB;MAAA,yC;MAkCA,8C;MAICA,wB;QA+CI,WAAW,8B;QACX,aAnC8C,KAmCj  
C,E;QAnCb,OAoCO,oBAAW,MAAX,EAAMb,IAAK,aAAxB,C;O;KAjDX,C;mGAgBA,yB;MAAA,8C;MAAA,m  
C;QAaI,WAAW,mB;QACX,aAAa,O;QACb,OAAO,oBAAW,MAAX,EAAMb,IAAK,aAAxB,C;O;KafX,C;mGak  
BA,yB;MAAA,8C;MAAA,mC;QAaI,WAAW,mB;QACX,aAAa,O;QACb,OAAO,oBAAW,MAAX,EAAMb,IAAK  
,aAAxB,C;O;KafX,C;IjK/CA,2E;MASI,sC;MAAA,4C;K;IATJ,mGAWY,Y;MAAQ,2B;KAXpB,E;IAAA,4DAaQ,  
kB;MACI,wBAAW,MAAX,C;K;IAdZ,wF;IkKewC,sC;MACpC,0B;K;;IAGJ,kC;MAUI,OAA2C,CAA3C,2BAA6B  
,uBAA7B,EAAoC,KAApC,CAA2C,e;K;IAE/C,8B;K;kDAuBI,4B;MASI,MAAM,qCAA8B,8CAA9B,C;K;;IAW4  
B,8C;MAGtC,6B;MAEmD,UAMX,M;MAPxC,kBACmD,mE;MAEnD,eAC0B,K;MAE1B,cACwC,kE;MAExC,gB  
ACmC,gB;K;iGAG/B,Y;MAAQ,0C;K;0DAEZ,kB;MACI,cAAY,I;MACZ,gBAAC,M;K;IAGsE,iG;MAAA,uB;QA  
ExE,Q;QAAZ,qCAAY,8D;QACZ,sCAAa,a;QAFb,OAGA,yB;O;K;2DAJJ,+B;MAAKD,OAAsC,wDAAtC,c;K;IAO  
yE,uH;MAAA,uB;QAExG,Q;QAaf,iBA Ae,8F;QACf,eAAK,2B;QAA6B,mC;QxM/FtB,gBAAT,Q;QwMoG0D,kB;  
QAjZD,sBAAsB,SAAK,W;QAC3B,IAAI,eAAa,eAAjB,C;UAEL,iC;UACA,mBAAY,oCAAwB,eAAxB,EAAyC,kE  
AAzC,C;;UAGZ,mBAAY,kE;;QAEhB,oBAAa,e;QAZjB,OAcA,yB;O;K;6DAfJ,0C;MAAQf,OAAsC,qEAAtC,c;K;  
IAqBzB,mI;MAAA,qB;QACxD,yCAAgB,uB;QAGhB,qCAAY,Y;QACZ,uCAAc,E;QACIB,W;O;K;iEATA,iC;MA  
GwB,wCAAa,mCAAb,EAAoC,kFAApC,C;K;mDAQxB,Y;MAMuB,UADC,MACD,EAIH,MAJG,EAaK,M;MAjB  
xB,OAAO,IAAP,C;QAEI,aAAa,IAAK,S;QACF,SAAL,IAAK,O;QAAL,mB;UACyB,gBAArB,0D;UnKtBhB,U;U  
ADP,yB;UmKuBe,OnKtBR,sF;;QmKqBC,WAAW,M;QAGX,IAAI,mDAAoB,MAApB,QA AJ,C;;YAliB,SAAT,ej  
KtJV,CiKsJuD,IjKtJvD,EiKsJ6D,YjKtJ7D,EiKsJoE,IjKtJpE,EAA8C,KAA9C,C;;YiKuJQ,gC;cACE,IiKvJhB,oBDg  
DQ,WAAO,cmKuG0B,CnKvG1B,CAAP,CChDR,C;ckKwJgB,Q;;cALI,O;;UAAR,c;UAQA,IAAI,MAAM,yBAAV  
,C;YACI,IiKrKhB,oBDgDQ,WmKqHoB,0EnKrHpB,CChDR,C;;UkKwKY,gBAAC,gB;UACd,IAAK,oBAAW,MA  
AX,C;;K;;0ECxMrB,4B;MAyLI,QApLK,SAoLG,GApLoB,KAOlpB,I;MACR,IAAI,CArLC,SAqLD,GArLwB,KA  
qLxB,IAAiB,CAAjB,IAAsB,eArLE,KaqLF,MarLrB,SAqLL,C;QAA6C,a;;MarL7C,OAsLoC,C;K;kEApLX,yB;M  
AAA,0B;MAAA,mC;QAgMI,QAvLK,SAuLG,GAvLe,KAuLf,I;QAvLR,OAAgC,OAwLzB,KAxLgB,KAwLX,GA  
AW,CAAC,CAAC,IAxLF,KAwLC,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,KAxL  
yB,C;O;KATpC,C;4EAWA,4B;MAuKI,QAIKK,SAkKG,GAIKoB,KAKpB,I;MACR,IAAI,CAnKC,SAmKD,GAn  
KwB,KAmKxB,IAAiB,CAAjB,IAAsB,eAnKE,KAmKF,ManKrB,SAmKL,C;QAA6C,a;;ManK7C,OAOkoC,C;K;k  
EAIKX,yB;MAAA,4B;MAAA,mC;QA8KI,QArKK,SAqKG,GArKe,KaqKf,I;QArKR,OAAgC,QAsKzB,KAtKgB,  
KAsKX,GA AW,CAAC,CAAC,IAtKF,KAsKC,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EA  
AID,KAtKyB,C;O;KATpC,C;4EAWA,4B;MAqJI,QAhJK,SAgJG,GAhJc,KAgJd,I;MACR,IAAI,CAjJC,SAiJD,GA  
jKb,KAiJIB,IAAiB,CAAjB,IAAsB,eAjJJ,KAiJI,MAjJrB,SAiJL,C;QAA6C,a;;MAjJ7C,OAkJO,C;K;kEAhJX,4B;M

A4JI,QAnJK,SAmJG,GAnJS,KAmJT,I;MAnJR,OAoJO,KApJU,KAoJL,GAAW,CAAC,CAAC,IAPJR,KAoJO,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,K;K;4EAlJX,yB;MA6NA,0B;MA7NA,mC;QAKkB,kBAAT,oBAAL,SAAK,C;QA6NL,QAAQ,gBA7Ne,KA6Nf,C;QACR,IAAI,gBA9NmB,KA8NnB,eAAiB,CAAjB,IAAsB,mBA9NH,KA8NG,GAAa,WAAb,CAA1B,C;UAA6C,W;;QA9N7C,OA+NO,C;O;KApOX,C;kEAOA,4B;MAyOI,QAhOK,oBAAL,SAAK,CAgOG,QAhOU,KAgOV,C;MAhOR,OAiOO,MAjOW,KAiON,KAAa,MAjOP,KAiOO,CAAD,KAAmB,KAAM,CAAD,aAAL,CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,C;K;4EA/NX,4B;MAiHI,QA5GK,SA4GG,GA5GoB,KA4GpB,I;MACR,IAAI,CA7GC,SA6GD,GA7GwB,KA6GxB,IAAiB,CAAjB,IAAsB,eA7GE,KA6GF,MA7GrB,SA6GL,C;QAA6C,a;;MA7G7C,OA8GO,C;K;kEA5GX,yB;MAAA,0B;MAAA,mC;QAwHI,QA/GK,SA+GG,GA/Ge,KA+Gf,I;QA/GR,OAAgC,OAgHzB,KAhHgB,KAgHX,GAAW,CAAC,CAAC,IAhHF,KAgHC,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,KAhHyB,C;O;KATpC,C;4EAWA,4B;MA+FI,QA1FK,SA0FG,GA1FoB,KA0FpB,I;MACR,IAAI,CA3FC,SA2FD,GA3FwB,KA2FxB,IAAiB,CAAjB,IAAsB,eA3FE,KA2FF,MA3FrB,SA2FL,C;QAA6C,a;;MA3F7C,OA4FO,C;K;kEA1FX,yB;MAAA,4B;MAAA,mC;QAsGI,QA7FK,SA6FG,GA7Fe,KA6Ff,I;QA7FR,OAAgC,QA8FzB,KA9FgB,KA8FX,GAAW,CAAC,CAAC,IA9FF,KA8FC,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,KA9FyB,C;O;KATpC,C;4EAWA,4B;MA6EI,QAxEK,SAwEG,GAXeC,KAwEd,I;MACR,IAAI,CAzEC,SAyED,GAzEkB,KAyEiB,IAAiB,CAAjB,IAAsB,eAzEJ,KAyEI,MAzErB,SAyEL,C;QAA6C,a;;MAzE7C,OA0EO,C;K;kEAxEX,4B;MAoFI,QA3EK,SA2EG,GA3ES,KA2ET,I;MA3ER,OA4EO,KA5EU,KA4EL,GAAW,CAAC,CAAC,IA5ER,KA4EO,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,K;K;4EA1EX,yB;MAqJA,0B;MARJA,mC;QAKkB,kBAAT,oBAAL,SAAK,C;QAqJL,QAAQ,gBArJe,KAqJf,C;QACR,IAAI,gBAtJmB,KAsJnB,eAAiB,CAAjB,IAAsB,mBAtJH,KAsJG,GAAa,WAAb,CAA1B,C;UAA6C,W;;QAtJ7C,OAuJO,C;O;KA5JX,C;kEAOA,4B;MAiKI,QAxJK,oBAAL,SAAK,CAwJG,QAxJU,KAwJV,C;MAxJR,OAyJO,MAzJW,KAyJN,KAAa,MAzJP,KAyJO,CAAD,KAAmB,KAAM,CAAD,aAAL,CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,C;K;2EAxJX,4B;MAyCI,QApCA,SAoCQ,GApCY,KAoCZ,I;MACR,IAAI,CArCJ,SAqCI,GArcgB,KAqChB,IAAiB,CAAjB,IAAsB,eArCN,KAqCM,MArC1B,SAqCA,C;QAA6C,a;;MArC7C,OAsCO,C;K;iEApCX,yB;MAAA,0B;MAAA,mC;QAgDI,QAvCA,SAuCCQ,GAvCO,KAuCP,I;QAvCR,OAAwB,OAwCjB,KAxCQ,KAwCH,GAAW,CAAC,CAAC,IAxCV,KAwCS,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,KAxCiB,C;O;KAT5B,C;4EAWA,4B;MAuBI,QAlBA,SAkBQ,GAIBY,KAkBZ,I;MACR,IAAI,CAnBJ,SAmBI,GAnBgB,KAmBhB,IAAiB,CAAjB,IAAsB,eAnBN,KAmBM,MAnB1B,SAmBA,C;QAA6C,a;;MAnB7C,OAoBO,C;K;mEAlBX,yB;MAAA,4B;MAAA,mC;QA8BI,QArBA,SAqBQ,GArcBO,KAqBP,I;QArBR,OAAwB,QAsBjB,KAtBQ,KAsBH,GAAW,CAAC,CAAC,IAtBV,KAsBS,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,KATBiB,C;O;KAT5B,C;4EAWA,4B;MAKI,QAAQ,YAAO,KAAP,I;MACR,IAAI,aAAS,KAAT,IAAiB,CAAjB,IAAsB,eAAI,KA AJ,MAAa,SAAvC,C;QAA6C,a;;MAC7C,OAAO,C;K;mEAGX,4B;MASI,QAAQ,YAAO,KAAP,I;MACR,OAAO,KAAK,QA AW,CAAC,CAAC,IAAM,KAAP,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,K;K;4EAGX,yB;MAwEA,0B;MAx EA,mC;QAKkB,kBAAT,oBAAL,SAAK,C;QAwEL,QAAQ,gBAxEe,KAwEf,C;QACR,IAAI,gBAzEmB,KAyEnB,eAAiB,CAAjB,IAAsB,mBAzEH,KAyEG,GAAa,WAAb,CAA1B,C;UAA6C,W;;QAzE7C,OA0EO,C;O;KA/EX,C;kEAOA,4B;MAoFI,QA3EK,oBAAL,SAAK,CA2EG,QA3EU,KA2EV,C;MA3ER,OA4EO,MA5EW,KA4EN,KAAa,MA5EP,KA4EO,CAAD,KAAmB,KAAM,CAAD,aAAL,CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,C;K;6EA1EX,yB;MAsDA,0B;MATDA,mC;QAKS,cAAe,oBAAN,KAAM,C;QAsDpB,QatDA,SA SDQ,KAAO,OAAP,C;QACR,IAvDA,SAuDI,KAAS,OAAT,eAAiB,CAAjB,IAAsB,mBAAL,OA AJ,GA vD1B,SAuD0B,CAA1B,C;UAA6C,W;;QAvD7C,OA wDO,C;O;KA7DX,C;mEAOA,yB;MAAA,0B;MAAA,mC;QASS,cAAU,oBAAN,KAA M,C;QAYDf,QAZDA,SAyDQ,QAAO,OAAP,C;QAZDR,OAAYB,OA0DIB,MAAK,YAAa,MAAM,OAAN,CAAD,KAAmB,KAAM,CAAD,aAAL,CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,CA1DkB,S;O;KAT7B,C;6EAWA,yB;MAoCA,0B;MApCA,mC;QAKS,cAAe,oBAAN,KAAM,C;QAoCpB,QApCA,SAoCQ,KAAO,OAAP,C;QACR,IArCA,SAqCI,KAAS,OAAT,eAAiB,CAAjB,IAAsB,mBAAL,OA AJ,GArc1B,SAqC0B,CAA1B,C;UAA6C,W;;QArC7C,OAsCO,C;O;KA3CX,C;mEAOA,yB;MAAA,4B;MAAA,mC;QASS,cAAU,oBAAN,KAAM,C;QAuCF,QAvCA,SAuCCQ,QAAO,OAAP,C;QAvCR,OAAYB,QAwCIB,MAAK,YAAa,MAAM,OAAN,CAAD,KAAmB,KAAM,CAAD,aAAL,CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,CAXCkB,S;O;KAT7B,C;6EAWA,yB;MAkBA,0B;MAlBA,mC;QAKS,cAAe,oBAAN,KAAM,C;QAKBpB,QAlBA,SAkBQ,KAAO,OAAP,C;QACR,IAnBA,SAmBI,KA

AS,OAAT,eAAiB,CAAjB,IAAsB,mBAAI,OAAJ,GAnB1B,SAmB0B,CAA1B,C;UAA6C,W;;QAnB7C,OAoBO,C; O;KazBX,C;mEAOA,4B;MASS,cAAU,oBAAN,KAAM,C;MAqBf,QArBA,SAqBQ,QAAO,OAAP,C;MarBR,OA sBO,MAAK,YAAa,MAAM,OAAN,CAAD,KAAMb,KAAM,CAAD,aAAL,CAAnB,CAAD,YAAkC,EAAIC,CAA X,CAAL,CatBkB,Q;K;6EAE7B,yB;MAAA,0B;MAAA,mC;QAKI,QAAQ,cAAO,KAAP,C;QACR,IAAI,cAAS,K AAT,eAAiB,CAAjB,IAAsB,mBAAI,KAAJ,GAAa,SAAb,CAA1B,C;UAA6C,W;;QAC7C,OAAO,C;O;KAPX,C;m EAUa,4B;MASI,QAAQ,iBAAO,KAAP,C;MACR,OAAO,MAAK,UAAa,MAAM,KAAN,CAAD,KAAMb,KAAM ,CAAD,aAAL,CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,C;K;kEAGX,yB;M1GmqB2C,0B;M0GnqB3C,mC; QAWI,QAAQ,YAAO,K;QACJ,iBAAS,G;QAAT,S;UAAsB,O1GupB0C,W0GvpB1C,C1GupB0C,C0GvpB1C,K1G upB0C,W0GvpBhC,K1GupBgC,C;;Q0GvpB3E,OAAO,OAAgD,IAAI,KAApD,GAA+D,C;O;KAZ1E,C;mEAEa,y B;M1G0H6C,0B;M0G1H7C,mC;QAqCI,QA1BK,SA0BG,GA1BY,K;QA2BT,iBAAK,G;QAAL,S;UAAy,O1GoF0 B,W0GpF1B,C1GoF0B,C0GpF1B,K1GoF0B,W0G/G7B,K1G+G6B,C;;Q0G/GjD,OA2BO,OAAc,IA3BzB,KA2B b,GAAqD,C;O;KAtChE,C;mEAAa,yB;M1G6G6C,0B;M0G7G7C,mC;QAwBI,QAbA,SAAQ,GAbO,K;QAcJ,iBAA K,G;QAAL,S;UAAy,O1GoF0B,W0GpF1B,C1GoF0B,C0GpF1B,K1GoF0B,W0GIGiC,K1GkGkC,C;;Q0GIGjD,O AcO,OAAc,IA9B,KAcR,GAAqD,C;O;KazBhE,C;mEAAa,yB;M1GgG6C,0B;M0GhG7C,mC;QAWI,QAAQ,Y AAO,K;QACJ,iBAAK,G;QAAL,S;UAAy,O1GoF0B,W0GpF1B,C1GoF0B,C0GpF1B,K1GoF0B,W0GpFhB,K1G oFgB,C;;Q0GpFjD,OAAO,OAAc,IAAI,KAA1C,GAAqD,C;O;KAZhE,C;4ECtVA,yB;MAAA,8B;MAAA,4B;QA OyC,Q;QAAA,gFAAoB,C;O;KAP7D,C;ICM0B,4C;MA+CtB,qC;MA/CuB,kB;MAAgB,kB;MAAgB,kB;MAMvD, iBAAsB,iBAAU,UAAV,EAAiB,UAAjB,EAAwB,UAAxB,C;K;0CAEtB,+B;M9MWA,IAAI,E8MViB,CAAT,sBA AY,GAAZ,KAA4C,CAAT,sBAAY,GAA/C,MAA+E,CAAT,sBAAY,GAAIF,C9MUR,CAAJ,C;QACI,c8MVI,2E; Q9MWJ,MAAM,gCAAYB,OAAQ,WAAjC,C;;M8MTN,OAAO,CAAA,KAAM,IAAI,EAAV,KAAgB,KAAM,IAA I,CAA1B,IAA+B,KAA/B,I;K;uCAGX,Y;MAGkC,OAAE,UAAF,oBAAS,UAAT,SAAgB,U;K;qCAEID,iB;MAEw B,gB;MADpB,IAAI,SAAS,KAAb,C;QAAoB,OAAO,I;MACP,iE;MAAD,mB;QAA6B,OAAO,K;;MAAvD,mBAA mB,M;MACnB,OAAO,IAAK,UAAAL,KAAGB,YAAa,U;K;uCAGx C,Y;MAA+B,qB;K;8CAE/B,iB;MAAoD,wBAA U,KAAM,UAAhB,I;K;gDAEpD,wB;MAKI,OAAA,IAAK,MAAL,GAAa,KAAb,KAAuB,IAAK,MAAL,KAAC,KA Ad,IACf,IAAK,MAAL,IAAc,KADtB,C;K;gDAGJ,+B;MAKI,OAAA,IAAK,MAAL,GAAa,KAAb,KAAuB,IAAK, MAAL,KAAC,KAAd,KACd,IAAK,MAAL,GAAa,KAAb,KAAsB,IAAK,MAAL,KAAC,KAAd,IACf,IAAK,MAAL ,IAAc,KADrB,CADc,CAAvB,C;K;IAIJ,mC;MAAA,uC;MACI,2BAIuC,G;MAEvC,eAIoC,uCAA0B,M;K;;;IAXIE, +C;MAAA,8C;QAAA,6B;;MAAA,uC;K;;IA9CA,iD;MAAA,uD;MAG6C,0BAAK,KAAL,EAAy,KAAZ,EAAmB, CAAnB,C;MAH7C,Y;K;IA6DJ,qC;MAAA,yC;K;8CAEI,Y;MAC2B,yBAAc,CAAd,EAAiB,CAAjB,EAAoB,EAA pB,C;K;;IAH/B,iD;MAAA,gD;QAAA,+B;;MAAA,yC;K;4FCxDI,yB;MAAA,2D;MAAA,4B;QAAQ,MAAM,6BA AoB,6BAApB,C;O;KAAd,C;;;ICSJ,uB;MAG2C,+BAAoB,KAApB,C;K;4EAE3C,wC;MAO4F,sB;K;IAE5F,6C;M AAA,e;MAAA,iB;MAAA,uB;K;IAAA,2C;MAAA,8C;O;MAKI,wF;MAKA,sF;MAMA,wE;K;;IAXA,yD;MAAA,i C;MAAA,iD;K;;IAKA,wD;MAAA,iC;MAAA,gD;K;;IAMA,iD;MAAA,iC;MAAA,yC;K;;IAhBJ,uC;MAAA,iJ;K;; IAAA,4C;MAAA,a;AAAA,c;UAAA,sD;aAAA,a;UAAA,qD;aAAA,M;UAAA,8C;;UAAA,gE;;K;;IAyBA,+B;MAA A,mC;K;;IAAA,2C;MAAA,0C;QAAA,yB;;MAAA,mC;K;IAGoC,qC;MACHC,qBAAsC,W;MACtC,gBAA2B,iC; K;uFAGvB,Y;MAMW,Q;MALP,IAAI,kBAAW,iCAAf,C;QACI,gBAAS,mC;QACT,qBAAc,I;;MAGIB,OAAO,gF; K;6CAGf,Y;MAAwC,yBAAW,iC;K;wCAEnD,Y;MAAkC,OAAI,oBAAJ,GAA2B,SAAN,UAAM,CAA3B,GAA2 C,iC;K;8CAE7E,Y;MAAkC,+BAAoB,UAApB,C;K;;IAGG,oC;MAAC,4B;K;wEAAA,Y;MAAA,2B;K;kDAEtC,Y; MAAwC,W;K;6CAExC,Y;MAAkC,OAAM,SAAN,UAAM,C;K;;oFC2C5C,yB;MAAA,gD;MAAA,4B;QAM6C,O AAmB,aAAIB,YAAy,GAAM,C;O;KANhE,C;oGAQA,yB;M9G7FA,4B;M8G6FA,4B;QAMqD,O9G7FM,Y8G6F L,YAAy,G9G7FP,C8G6FN,GAA6C,EAA7C,I;O;KANrD,C;sGAQA,yB;MAAA,kE;MAAA,4B;QAMsD,OAAmB, sBAAIB,YAAW,GAAO,C;O;KANzE,C;8FAQA,yB;MAAA,0D;MAAA,0B;MAAA,4B;QAOMD,OAAuC,OAApB ,kBAAIB,YAAy,GAAM,CAAoB,C;O;KAP1F,C;4FASA,yB;MAAA,wD;MAAA,0B;MAAA,4B;QAOKD,OAA2B, OAAAnB,iBAAR,SAAQ,CAAmB,C;O;KAP7E,C;IAUA,2C;MAaI,OAA+E,OAA9E,SAAQ,KAAI,WAAa,CAAjB,C AAR,GAAkD,CAAIB,YAAy,GAAM,MAAK,CAAL,IAAU,WAAa,CAAvB,CAA4B,C;K;IAEnF,4C;MAaI,OAA+ E,OAA9E,SAAQ,IAAI,CAAJ,IAAS,WAAa,CAAtB,CAAR,GAAwD,CAAIB,YAAy,GAAM,OAAK,WAAa,CAA I B,CAAsB,C;K;oFAEnF,yB;MAAA,gD;MAAA,4B;QAM8C,OAAqB,aAApB,YAAy,KAAQ,C;O;KANnE,C;oGA QA,yB;M9GtKA,4B;M8GsKA,4B;QAOI,O9GvKuD,Y8GuKtD,YAAy,K9GvK0C,C8GuKvD,GAA+C,EAA/C,I;O

;KAPJ,C;sGASA,yB;MAAA,kE;MAAA,4B;QAMuD,OAAqB,sBAApB,YAAW,KAAS,C;O;KAN5E,C;8FAQA,y  
B;MAAA,0D;MAAA,4B;MAAA,4B;QAOqD,OAAyC,QAAPB,kBAAPB,YAAy,KAAQ,CAAoB,C;O;KAP9F,C;4  
FASA,yB;MAAA,wD;MAAA,4B;MAAA,4B;QAOoD,OAA2B,QAAnB,iBAAR,SAAQ,CAAmB,C;O;KAP/E,C;IA  
UA,2C;MAaI,OAAoF,QAAnF,SAAQ,KAAI,WAAa,EAAjB,CAAR,GAAqD,CAAPB,YAAy,KAAQ,MAAK,EAA  
L,IAAW,WAAa,EAAxB,CAA8B,C;K;IAExF,4C;MAaI,OAAoF,QAAnF,SAAQ,IAAI,EAAJ,IAAU,WAAa,EAAv  
B,CAAR,GAA4D,CAAPB,YAAy,KAAQ,OAAK,WAAa,EAAIB,CAAU,B,C;K;0EjNIRxF,yB;MAaA,kF;MAbA,w  
B;QAUbI,IAAI,CAbI,KAAr,C;UACI,cAda,qB;UAeb,MAAM,8BAAYB,OAAQ,WAAjC,C;;O;KAZbD,C;0EAaA,y  
B;MAAA,kF;MAAA,qC;QAU,IAAI,CAAC,KAAL,C;UACI,cAAc,a;UACd,MAAM,8BAAYB,OAAQ,WAAjC,C;;  
O;KAZd,C;sFAgBA,yB;MAWA,kF;MAXA,wB;QAQW,yB;QAeP,IAfsB,KAelB,QAAJ,C;UACI,cAhB2B,0B;UAI  
B3B,MAAM,8BAAYB,OAAQ,WAAjC,C;;UAEN,wBAnBkB,K;;QAAtB,4B;O;KARJ,C;wFAWA,yB;MAAA,kF;  
MAAA,qC;QAYI,IAAI,aAAJ,C;UACI,cAAc,a;UACd,MAAM,8BAAYB,OAAQ,WAAjC,C;;UAEN,OAAO,K;;O;K  
AhBf,C;oEAoBA,yB;MAaA,4E;MAbA,wB;QAUbI,IAAI,CAbE,KAAr,C;UACI,cAdW,e;UAeX,MAAM,2BAAsB,  
OAAQ,WAA9B,C;;O;KAZbD,C;sEAaA,yB;MAAA,4E;MAAA,qC;QAU,IAAI,CAAC,KAAL,C;UACI,cAAc,a;U  
ACd,MAAM,2BAAsB,OAAQ,WAA9B,C;;O;KAZd,C;kFAgBA,yB;MAcA,4E;MAdA,wB;QAWW,uB;QAeP,IAfo  
B,KAehB,QAAJ,C;UACI,cAhByB,0B;UAIbZB,MAAM,2BAAsB,OAAQ,WAA9B,C;;UAEN,sBAnBgB,K;;QAAP  
B,0B;O;KAXJ,C;oFAcA,yB;MAAA,4E;MAAA,qC;QAYI,IAAI,aAAJ,C;UACI,cAAc,a;UACd,MAAM,2BAAsB,O  
AAQ,WAA9B,C;;UAEN,OAAO,K;;O;KAhBf,C;oEAqBA,yB;MAAA,4E;MAAA,0B;QAMiD,MAAM,2BAAsB,O  
AAQ,WAA9B,C;O;KANvD,C;IwCnHiC,uB;MA2D7B,8B;MA1DA,kB;K;mFAS8B,Y;MAAQ,iD;K;mFAMR,Y;M  
AAQ,gD;K;wFAItC,yB;MAAA,gB;MAAA,8B;MAAA,mB;QAWgB,Q;QADR,mB;UADJ,OACiB,I;;UADjB,OAE  
Y,2E;O;KAXhB,C;uCACa,Y;MAQQ,kBADE,UACF,kB;QADJ,OACkB,UAAM,U;;QADxB,OAEY,I;K;gCAGhB,  
Y;MAOQ,kBADE,UACF,kB;QADJ,OACkB,UAAM,W;;QADxB,OAEY,sBAAU,UAAV,O;K;IAKhB,4B;MAAA,  
gC;K;wHAKI,yB;MAAA,iC;MAAA,wB;QAOI,uBAAO,KAAP,C;O;KAPJ,C;wHASA,yB;MAAA,kD;MAAA,iC;  
MAAA,4B;QAOI,uBAAO,cAAc,SAAd,CAAP,C;O;KAPJ,C;;IADJ,wC;MAAA,uC;QAAA,sB;;MAAA,gC;K;IAW  
BsB,mC;MACIB,0B;K;sCAGA,iB;MAA4C,+CAAoB,uBAAa,KAAM,UAAAnB,C;K;wCACHe,Y;MAA+B,OAAU,  
SAAV,cAAU,C;K;wCACzC,Y;MAAkC,oBAAU,cAAV,M;K;;gCA/FIC,Y;MAAA,c;MAOI,sD;MAPJ,a;K;8BA  
AA,iB;MAAA,2IAOI,sCAPJ,G;K;IAmGA,kC;MAOI,OAAO,mBAAQ,SAAR,C;K;IAEX,mC;MAQI,IAAI,8CAAJ,  
C;QAA6B,MAAM,eAAM,U;K;gFAG7C,yB;MAAA,4B;MAAA,qB;MAxQC,kD;MAwCR,wB;QAOW,Q;;UACI,O  
AIDH,WAKDW,OAIDX,C;;UAmDN,gC;YACS,OA3CH,WAAO,cA2CI,CA3CJ,CAAP,C;;YAwCD,O;;QAAP,W;  
O;KAPJ,C;kFAcA,yB;MAAA,4B;MAAA,qB;MAtdQ,kD;MAsDR,mC;QAOW,Q;;UACI,OAhEH,WAgEW,gBAh  
EX,C;;UAiEN,gC;YACS,OAzDH,WAAO,cAyDI,CAzDJ,CAAP,C;;YAsDD,O;;QAAP,W;O;KAPJ,C;8EAgBA,yB;  
MAAA,oD;MAAA,gB;MAAA,8B;MAAA,4B;QAUW,Q;QADP,yB;QACA,OAAO,gF;O;KAVX,C;+EAaA,yB;M  
AAA,gB;MAAA,8B;MAAA,uC;QAegB,UADL,M;QAAM,gBAAGB,2B;QACzB,sB;UAAQ,yF;;UACA,mBAAU,S  
AAV,C;QAFZ,a;O;KAdJ,C;kFAoBA,yB;MAAA,gB;MAAA,8B;MAAA,0C;QAUW,Q;QADP,IAAI,mBAAJ,C;UA  
Ae,OAAO,Y;QActB,OAAO,gF;O;KAVX,C;qEAaA,yB;MAAA,gB;MAAA,8B;MAAA,kD;QAIb0B,UAdf,M;QA  
AM,gBAAGB,2B;QACzB,sB;UAAQ,mBAAU,gFAAV,C;;UACA,mBAAU,SAAV,C;QAFZ,a;O;KAhBJ,C;mEAw  
BA,yB;MAAA,4B;MAAA,gB;MAAA,8B;MAAA,uC;YAe8C,I;YADnC,M;QACH,wB;UAAa,gB;UAAO,SA7JhB,  
WA6JwB,UAAU,gFAAV,CA7JxB,C;;UA8JI,oBAAO,eAAP,C;QAFZ,a;O;KAdJ,C;gFAoBA,yB;MAAA,gB;MAA  
A,8B;MAAA,iC;MA1GA,qB;MAtdQ,kD;MAgKR,uC;QAWW,Q;QACH,wB;UA/GG,U;;YA+GkC,U;YA9G9B,S  
AhEH,gBA8KuB,UAAU,sFAAV,CA9KvB,C;;YAIEEN,gC;cACS,SAzDH,gBAAO,cAyDI,CAzDJ,CAAP,C;;cAsDD  
,O;;UA+GU,a;;UACL,uBAAO,eAAP,C;QAFZ,W;O;KAXJ,C;wEAIbA,yB;MAAA,4B;MAAA,uC;QAcW,Q;QAA  
M,gBAAGB,2B;QACzB,sB;UAAQ,gB;;UACO,OAnMX,WAmMmB,UAAU,SAAV,CAnMnB,C;;QAIrMR,W;O;K  
AdJ,C;wFAoBA,yB;MA/IA,4B;MAAA,qB;MAtdQ,kD;MAqMR,uC;QAWW,Q;QAAM,gBAAGB,2B;QACzB,sB;  
UAAQ,gB;;UApJL,U;;YACI,SAhEH,WAOmNkB,oBApNIB,C;;YAIEEN,gC;cACS,SAzDH,WAAO,cAyDI,CAzDJ,C  
AAP,C;;cAsDD,O;;UAqJK,a;;QAFZ,W;O;KAXJ,C;4EAmBA,6B;MAUI,Q;MAAA,iD;QAAyB,Y;;MACzB,OAAO  
,S;K;4EAGX,yB;MAAA,gB;MAAA,8B;MAAA,oC;QAU0B,Q;QAAtB,IAAI,mBAAJ,C;UAAe,OAAO,gFAAP,C;;  
QACf,OAAO,S;O;KAXX,C;IrCtTgC,sC;MAAC,uB;QAAA,UAAkB,kC;mBAA4C,O;;K;;0DAE/F,yB;MAAA,2D;  
MAAA,mB;QAKoC,MAAM,8B;O;KAL1C,C;oEAOA,yB;MAAA,2D;MAAA,yB;QAMkD,MAAM,6BAAoB,sCA  
AmC,MAAvD,C;O;KANxD,C;gEAUA,iB;MAUI,OAAO,O;K;KEAGX,4B;MAUI,OAAO,gB;K;oEAGX,2B;MAUI



,OAAgB,MAAT,QAAS,C;K;oEAGpB,4B;MAUI,gB;MACA,OAAO,S;K;kEAGX,4B;MAWI,MAAM,SAAN,C;M  
ACA,OAAO,S;K;kEAGX,4B;MAUI,OAAO,MAAM,SAAN,C;K;sEAGX,gC;MAWI,OAAW,UAAU,SAAV,CAAJ  
,GAAqB,SAArB,GAA+B,I;K;8EAG1C,gC;MAWI,OAAW,CAAC,UAAU,SAAV,CAAL,GAAsB,SAAtB,GAAGC,  
I;K;wEAG3C,yB;MAWI,iBAAc,CAAd,UAAsB,KAAtB,U;QACI,OAAO,KAAP,C;;K;wE+MjJR,iB;MAGkF,Y;K;I  
Ca9C,6B;MACHC,kB;MACA,oB;K;8BAGA,Y;MAGyC,aAAG,UAAH,UAAW,WAAX,M;K;;gCAvB7C,Y;MAGB  
I,iB;K;gCAhBJ,Y;MAiBi,kB;K;kCAjBJ,yB;MAAA,gBAGBI,qCAhBJ,EAiBI,wCAjBJ,C;K;8BAAA,Y;MAAA,c;M  
AgBI,sD;MACA,uD;MAjBJ,a;K;4BAAA,iB;MAAA,4IAGBI,sCAhBJ,IAiBI,wCAjBJ,I;K;IA0BA,6B;MAMoD,gB  
AAK,SAAL,EAAW,IAAX,C;K;IAEpD,8B;MAI8C,iBAAO,eAAP,EAAC,gBAAd,E;K;IAiBD,sC;MACzC,kB;MA  
CA,oB;MACA,kB;K;gCAGA,Y;MAGyC,aAAG,UAAH,UAAW,WAAX,UAAoB,UAApB,M;K;;kCAx7C,Y;MA  
gBI,iB;K;kCAhBJ,Y;MAiBi,kB;K;kCAjBJ,Y;MAkBI,iB;K;oCAIBJ,gC;MAAA,kBAGBI,qCAhBJ,EAiBI,wCAjBJ,  
EAkBI,qCAIBJ,C;K;gCAA,Y;MAAA,c;MAGBI,sD;MACA,uD;MACA,sD;MAIBJ,a;K;8BAAA,iB;MAAA,4IAG  
BI,sCAhBJ,IAiBI,wCAjBJ,IAkBI,sCAIBJ,I;K;IA2BA,8B;MAImD,iBAAO,eAAP,EAAC,gBAAd,EAAsB,eAAtB,E;  
K;IhOIE1B,qB;MAErB,6B;MAFkG,gB;K;IAEIG,2B;MAAA,+B;MACI,iBAGoC,UAAM,CAAN,C;MAEpC,iBAG  
oC,UAAM,MAAN,C;MAEpC,kBAGmC,C;MAEnC,iBAGkC,C;K;;IANtC,uC;MAAA,sC;QAAA,qB;;MAAA,+B  
;K;kGAsBA,iB;MAOmE,OAAa,0BA2O1C,SAAL,GAAiB,GA3O8B,EAAU,KA2OpD,KAAL,GAAiB,GA3O8B,C;  
K;sGAehF,iB;MAM2D,OAAa,0BAmO1C,SAAL,GAAiB,GAnOsB,EAAU,KEoO5C,KAAL,GAAiB,KFpOsB,C;K;  
sGAExE,yB;MA0PA,6B;MC3PA,8C;MDCA,wB;QAMyD,OCAS,YAAiB,CD6PhD,cAAU,SAAL,GAAiB,GAAtB,  
CC7PgD,MAAjB,EDAe,KCAc,KAA7B,C;O;KDNIE,C;sGAQA,yB;MA4PA,WAS6D,wB;MAT7D,+B;MkB7PA,g  
D;MIBCA,wB;QAM0D,OkBAS,aAakB,CIB+PhD,eAAW,oBAAL,SAAK,CAAL,UAAN,CkB/PgD,MAAiB,ElBA  
gB,KkBAc,KAA9B,C;O;KIBNnE,C;4FAQA,yB;MA0OA,6B;MA1OA,wB;QAEsD,OCMD,cAAU,CD2O5B,cAAU  
,SAAL,GAAiB,GAAtB,CC3O4B,MAAK,GAAW,CD2O5C,cAjPsC,KAIp5B,KAAL,GAAiB,GAAtB,CC3O4C,M  
AAX,IAAf,C;O;KDRrD,C;4FAGA,yB;MAuOA,6B;MAvOA,wB;QAEuD,OCGF,cAAU,CD2O5B,cAAU,SAAL,G  
AAiB,GAAtB,CC3O4B,MAAK,GAAW,CC4O5C,cF/OuC,KE+O7B,KAAL,GAAiB,KAAtB,CD5O4C,MAAX,IA  
Af,C;O;KDLrD,C;4FAGA,yB;MAoOA,6B;MApOA,wB;QAEqD,OCAA,cAAU,CD2O5B,cAAU,SAAL,GAAiB,G  
AAAtB,CC3O4B,MAAK,GDAI,KCAO,KAAZ,IAAf,C;O;KDFrD,C;4FAGA,yB;MA2OA,WAS6D,wB;MAT7D,+B;  
MA3OA,wB;QAEuD,OkBAA,eAAW,CIBkP7B,eAAW,oBAAL,SAAK,CAAL,UAAN,CkBIP6B,MAAK,KIBAI,K  
kBAO,KAAZ,CAAhB,C;O;KIBFvD,C;8FAIA,yB;MA6NA,6B;MA7NA,wB;QAEuD,OCMD,cAAU,CD8N7B,cA  
AU,SAAL,GAAiB,GAAtB,CC9N6B,MAAK,GAAY,CD8N9C,cApOwC,KAO09B,KAAL,GAAiB,GAAtB,CC9N8  
C,MAAZ,IAAf,C;O;KDRtD,C;8FAGA,yB;MA0NA,6B;MA1NA,wB;QAEwD,OCGF,cAAU,CD8N7B,cAAU,SA  
L,GAAiB,GAAtB,CC9N6B,MAAK,GAAY,CC+N9C,cFIOyC,KEo/B,KAAL,GAAiB,KAAtB,CD/N8C,MAAZ,I  
AAf,C;O;KDLtD,C;8FAGA,yB;MAuNA,6B;MAvNA,wB;QAEsD,OCAA,cAAU,CD8N7B,cAAU,SAAL,GAAiB,  
GAAtB,CC9N6B,MAAK,GDAK,KCAO,KAAZ,IAAf,C;O;KDFtD,C;8FAGA,yB;MA8NA,WAS6D,wB;MAT7D,+  
B;MA9NA,wB;QAEwD,OkBAA,eAAW,CIBqO9B,eAAW,oBAAL,SAAK,CAAL,UAAN,CkBrO8B,MAAK,UIBA  
K,KkBAO,KAAZ,CAAhB,C;O;KIBFxD,C;8FAIA,yB;MAGNA,6B;MAhNA,wB;QAEuD,OCMD,cAAe,YAAL,CD  
iN7B,cAAU,SAAL,GAAiB,GAAtB,CCjN6B,MAAK,EAAY,CDiN9C,cAvNwC,KAuN9B,KAAL,GAAiB,GAAtB,  
CCjN8C,MAAZ,CAAf,C;O;KDRtD,C;8FAGA,yB;MA6MA,6B;MA7MA,wB;QAEwD,OCGF,cAAe,YAAL,CDiN  
7B,cAAU,SAAL,GAAiB,GAAtB,CCjN6B,MAAK,EAAY,CCKn9C,cFrNyC,KEqN/B,KAAL,GAAiB,KAAtB,CDI  
N8C,MAAZ,CAAf,C;O;KDLtD,C;8FAGA,yB;MA0MA,6B;MA1MA,wB;QAEsD,OCAA,cAAe,YAAL,CDiN7B,c  
AAU,SAAL,GAAiB,GAAtB,CCjN6B,MAAK,EDAK,KCAO,KAAZ,CAAf,C;O;KDFtD,C;8FAGA,yB;MAiNA,W  
AS6D,wB;MAT7D,+B;MAjNA,wB;QAEwD,OkBAA,eAAW,CIBwN9B,eAAW,oBAAL,SAAK,CAAL,UAAN,Ck  
BxN8B,MAAK,UIBAK,KkBAO,KAAZ,CAAhB,C;O;KIBFxD,C;0FAIA,yB;MAmMA,6B;MC7LA,4C;MDNA,wB  
;QAEqD,OCMD,WDoMjB,cAAU,SAAL,GAAiB,GAAtB,CCpMiB,EDoMjB,cA1MoC,KA0M1B,KAAL,GAAiB,  
GAAtB,CCpMiB,C;O;KDRpD,C;0FAGA,yB;MAGMA,6B;MC7LA,4C;MDHA,wB;QAEsD,OCGF,WDoMjB,cAA  
U,SAAL,GAAiB,GAAtB,CCpMiB,ECqMjB,cFxMqC,KEwM3B,KAAL,GAAiB,KAAtB,CDrMiB,C;O;KDLpD,C;  
0FAGA,yB;MA6LA,6B;MC7LA,4C;MDAA,wB;QAEoD,OCAA,WDoMjB,cAAU,SAAL,GAAiB,GAAtB,CCpMi  
B,EDAKB,KCAIB,C;O;KDFpD,C;0FAGA,yB;MAoMA,WAS6D,wB;MAT7D,+B;MkBPMA,8C;MIBAA,wB;QAE  
sD,OkBAA,YIB2MjB,eAAW,oBAAL,SAAK,CAAL,UAAN,CkB3MiB,ElBAmB,KkBAnB,C;O;KIBftD,C;0FAIA,  
yB;MA5LA,6B;MCxKA,kD;MDdA,wB;QAMqD,OCcD,cD2KjB,cAAU,SAAL,GAAiB,GAAtB,CC3KiB,ED2KjB,

cAzLoC,KAyL1B,KAAL,GAAiB,GAAtB,CC3KiB,C;O;KDpBpD,C;0FAOA,yB;MA+KA,6B;MCxKA,kD;MDPA,wB;QAMsD,OCOF,cD2KjB,cAAU,SAAL,GAAiB,GAAtB,CC3KiB,EC4KjB,cFnLqC,KEmL3B,KAAL,GAAiB,KAAAtB,CD5KiB,C;O;KDbpD,C;0FAOA,yB;MAwKA,6B;MCxKA,kD;MDAA,wB;QAMoD,OCAA,cD2KjB,cAAU,SAAL,GAAiB,GAAtB,CC3KiB,EDAkB,KCAiB,C;O;KDNpD,C;0FAOA,yB;MA2KA,WAS6D,wB;MAT7D,+B;MkB3KA,oD;MIBAA,wB;QAMsD,OkBAA,eIB8KjB,eAAW,oBAAL,SAAK,CAAL,UAAN,CkB9KiB,EIBAmB,KkBAAnB,C;O;KIBNtD,C;oGAQA,yB;MAyJA,6B;MC7LA,4C;MDoCA,wB;QAMiD,OCxCG,WDoMjB,cAAU,SAAAL,GAAiB,GAAtB,CCpMiB,EDoMjB,cA5JqC,KA4J3B,KAAL,GAAiB,GAAtB,CCpMiB,C;O;KDKCpD,C;oGAOA,yB;MakJA,6B;MC7LA,4C;MD2CA,wB;QAMkD,OC/CE,WDoMjB,cAAU,SAAL,GAAiB,GAAtB,CCpMiB,ECqMjB,cFtJsC,KEsJ5B,KAAL,GAAiB,KAAAtB,CDrMiB,C;O;KDyCpD,C;oGAOA,yB;MA2IA,6B;MC7LA,4C;MDkDA,wB;QAMgD,OCtDI,WDoMjB,cAAU,SAAL,GAAiB,GAAtB,CCpMiB,EDsDmB,KCtDnB,C;O;KDGpD,C;oGAOA,yB;MA8IA,WAS6D,wB;MAT7D,+B;MkBPMA,8C;MIBsDA,wB;QAMkD,OkB1DI,YIB2MjB,eAAW,oBAAL,SAAK,CAAL,UAAN,CkB3MiB,EIB0DoB,KkB1DpB,C;O;KIBoDtD,C;0FAQA,yB;MA4HA,6B;MCxKA,kD;MDuOJ,0B;MAAA,+B;MA3LI,wB;QAQ6C,OA8LR,eAAW,OC5OI,cD2KjB,cAAU,SAAL,GAAiB,GAAtB,CC3KiB,ED2KjB,cA7H4B,KA6HIB,KAAL,GAAiB,GAAtB,CC3KiB,CakLf,KD0DW,CAAX,C;O;KATMrC,C;0FASA,yB;MAMHA,6B;MCxKA,kD;MCwOJ,4B;MAAA,iC;MFnLI,wB;QAQ+C,OEslR,gBAAY,QD7OC,cD2KjB,cAAU,SAAL,GAAiB,GAAtB,CC3KiB,EC4KjB,cFrH8B,KEqHpB,KAAL,GAAiB,KAAAtB,CD5KiB,CA4Lb,KCiDY,CAAZ,C;O;KF9LvC,C;0FASA,yB;MA0GA,6B;MCxKA,kD;MD8DA,wB;QAQ2C,OChES,cD2KjB,cAAU,SAAL,GAAiB,GAAtB,CC3KiB,EDgES,KChET,C;O;KDWpD,C;0FASA,yB;MA2GA,WAS6D,wB;MAT7D,+B;MkB3KA,oD;MIBgEA,wB;QAQ6C,OkBIES,eIB8KjB,eAAW,oBAAL,SAAK,CAAL,UAAN,CkB9KiB,EIBkEU,KkBIEV,C;O;KIB0DtD,C;0EAUA,yB;MAAA,0B;MAAA,+B;MAAA,mB;QAM0C,sBAAW,OAAL,SAAK,KAAX,C;O;KAN1C,C;0EAQA,yB;MAAA,0B;MAAA,+B;MAAA,mB;QAM0C,sBAAW,OAAL,SAAK,KAAX,C;O;KAN1C,C;kGAQA,yB;MAAA,8C;MAuEA,6B;MAvEA,wB;QAE8D,0BA8E3B,cAAU,SAAL,GAAiB,GAAtB,CA9E2B,EA8E3B,cA9EoD,KA8E1C,KAAL,GAAiB,GAAtB,CA9E2B,C;O;KAF9D,C;0FAIA,yB;MAAA,+B;M2LxOJ,0B;M3LwOI,wB;QAEEmD,sB2LvOgC,O3LuO1B,IAAK,K2LvOX,G3LuOoB,KAAM,K2LvOM,C3LuOhC,C;O;KAFnD,C;wFAGA,yB;MAAA,+B;M2LtoJ,0B;M3LsOI,wB;QAEkD,sB2Lro+B,O3LqOzB,IAAK,K2LrOX,G3LqOmB,KAAM,K2LrOM,C3LqO/B,C;O;KAFID,C;0FAGA,yB;MAAA,+B;M2LpOJ,0B;M3LoOI,wB;QAEEmD,sB2LnOgC,O3LmO1B,IAAK,K2LnOX,G3LmOoB,KAAM,K2LnOM,C3LmOhC,C;O;KAFnD,C;0EAGA,yB;MAAA,+B;M2LlOJ,0B;M3LkOI,mB;QAEiC,sB2LjOqB,OAAP,C3LiOR,S2LjOe,C3LiOrB,C;O;KAFjC,C;gFAIA,Y;MASmC,gB;K;kFACnC,yB;M2L1OJ,4B;M3L0OI,mB;QASqC,O2LhPiD,Q3LgP5C,S2LhPY,G3LgPE,G2LhP8B,C;O;K3LuOtF,C;8EAUA,Y;MASiC,OAAK,SAAL,GAAiB,G;K;gFACID,yB;MAAA,WASqD,wB;MATrD,mB;QASmC,OAAK,oBAAL,SAAK,CAAL,U;O;KATnC,C;kFAWA,Y;MAEqC,W;K;oFACrC,yB;MAAA,iC;M2L5QJ,4B;M3L4QI,mB;QASuC,uB2LlR+C,Q3LkRnC,S2LlRG,G3LkRW,G2LlRqB,C3LkR/C,C;O;KATvC,C;gFAUA,yB;MAAA,6B;MAAA,mB;QASmC,qBAAU,SAAL,GAAiB,GAAtB,C;O;KATnC,C;kFAUA,yB;MAAA,WAS6D,wB;MAT7D,+B;MAAA,mB;QASqC,sBAAW,oBAAL,SAAK,CAAL,UAAN,C;O;KATrC,C;kFAWA,Y;MAMqC,OApDC,SAAL,GAAiB,G;K;oFAqDID,Y;MAMuC,OA3DD,SAAL,GAAiB,G;K;+BA6DID,Y;MAAyC,OAAQ,CA7DX,SAAL,GAAiB,GA6DD,Y;K;+BA1UrD,Y;MAAA,c;MAGsG,qD;MAHtG,a;K;6BAAA,iB;MAAA,2IAGsG,oCAHtG,G;K;wEA8UA,yB;MAAA,+B;MAAA,4B;QAU0C,sBAAM,SAAN,C;O;KAV1C,C;0EAWA,yB;MAAA,0B;MAAA,+B;MAAA,4B;QAW2C,sBAAW,OAAL,SAAK,CAAX,C;O;KAX3C,C;0EAYA,yB;MAAA,0B;MAAA,+B;MAAA,4B;QAWyC,sBAAW,OAAL,SAAK,CAAX,C;O;KAXzC,C;0EAYA,yB;MAAA,0B;MAAA,+B;MAAA,4B;QAW0C,sBAAW,OAAL,SAAK,SAAX,C;O;KAX1C,C;IiC9WA,6B;MACqB,sB;K;uCAKjB,iB;MAM6C,OjCyUP,UiCzUO,aAAQ,KAAR,CjCyUP,C;K;uCiCvUtC,wB;MAOI,aAAQ,KAAR,IAAiB,KjCiOc,K;K;kFiC7NL,Y;MAAQ,OAAA,YAAQ,O;K;oCAE9C,Y;MAC8E,+BAAS,YAAT,C;K;IAExD,oC;MAAC,oB;MACnB,eAAoB,C;K;4CACpB,Y;MAAyB,sBAAQ,YAAM,O;K;yCACvC,Y;MAAoD,Q;MAA9B,IAAI,eAAQ,YAAM,OAAIB,C;QAAA,OjCoTY,UiCpTY,aAAM,mBAAN,EAAM,2BAAN,OjCoTZ,C;;QiCpT0C,MAAM,2BAAuB,YAAM,WAA7B,C;K;;0CAGtF,mB;MAIS,Q;MAALL,IAAI,eAAC,0EAAD,QAAJ,C;QAAiC,OAAO,K;MAExC,OAAe,WAAR,YAAQ,EAAS,OjC4MO,KiC5MhB,C;K;+CAGnB,oB;MACY,Q;MAA2B,gBAA3B,gE;MAA2B,c;;Qd6nDvB,U;QADhB,IAAI,wCAAsB,mBAA1B,C;UAaqC,aAAO,I;UAAP,e;;QACrB,6B;QAAhB,OAAgB,gBAAhB,C;UAAGB,2B;Uc7nD6B,2Bd6nDR,OC7nDQ,Q;UAAA,W;YAAuB,oBAAR,YAAQ,Ed6nD/B,OnBr7CF,KiCxmC,C;;Ud6nD9C,IAAI,OAAJ,C;YAAyB,aAAO,K;YA

AP,e;;;QAC/C,aAAO,I;;;Mc9nDH,iB;K;mCAGJ,Y;MAAkC,OAAA,IAAK,QAAQ,OAAb,KAAqB,C;K;;IA9CvD,s  
C;MAAA,oD;MACgC,uBAAK,cAAU,IAAV,CAAL,C;MADhC,Y;K;;;oCAPJ,Y;MAAA,OAKqB,qDALrB,M;K;o  
CAAA,Y;MAAA,c;MAKqB,wD;MALrB,a;K;kCAAA,iB;MAAA,2IAKqB,0CALrB,G;K;gFAwDA,yB;MAAA,yC;  
MAWsC,yC;QAAA,wB;UAAW,OAAA,aAAK,KAAL,CjCuLV,K;S;O;MiCIMvC,6B;QAWI,OAAO,oBAAW,+BA  
AU,IAAV,GAAgB,uBAAhB,CAAX,C;O;KAXX,C;kFAcA,oB;MAGqE,e;K;IhCrE7C,oB;MAEpB,4B;MAFiG,gB;  
K;IAEjG,0B;MAAA,8B;MACI,iBAGmC,SAAK,CAAL,C;MAEnC,iBAGmC,SAAK,EAAL,C;MAEnC,kBAGmC,  
C;MAEnC,iBAGkC,E;K;;;IANBtC,sC;MAAA,qC;QAAA,oB;;MAAA,8B;K;oGAsBA,yB;MD2QA,6B;MC3PA,8C;  
MAhBA,wB;QAM0D,OAIbQ,YAAy,IAAK,KAAjB,EAA6B,CD6P5D,cC9QsC,KD8Q5B,KAAL,GAAiB,GAAtB,  
CC7P4D,MAA7B,C;O;KAvBIE,C;oGAQA,yB;MCoQA,6B;MD5PA,8C;MARA,wB;QAM2D,OASO,YAAy,IAA  
K,KAAjB,EAA6B,CC8P5D,cDvQuC,KCuQ7B,KAAL,GAAiB,KAAtB,CD9P4D,MAA7B,C;O;KAFIE,C;gGAQA,  
yB;MAAA,8C;MAAA,wB;QAOKE,mBAAy,IAAK,KAAjB,EAAuB,KAAM,KAA7B,C;O;KAPIE,C;oGASA,yB;  
MAGRA,kBAS6D,sB;MAT7D,+B;MiBjRA,gD;MjBCA,wB;QAM0D,OiBAS,aAAkB,CjBmRhD,eAAW,oBAAL,S  
AAK,CAAL,iBAAN,CiBnRgD,MAAiB,EjBAgB,KiBAc,KAA9B,C;O;KjBNnE,C;0FAQA,yB;MD0OA,6B;MC1O  
A,wB;QAEsD,OAMD,cAAK,IAAK,KAAK,GAAW,CD2O5C,cCjP6B,KDiPnB,KAAL,GAAiB,GAAtB,CC3O4C,  
MAAX,IAAf,C;O;KARrD,C;0FAGA,yB;MCwOA,6B;MDxOA,wB;QAEuD,OAGF,cAAK,IAAK,KAAK,GAAW,  
CC4O5C,cD/O8B,KC+OpB,KAAL,GAAiB,KAAtB,CD5O4C,MAAX,IAAf,C;O;KALrD,C;0FAGA,yB;MAAA,6B  
;MAAA,wB;QAEqD,qBAAK,IAAK,KAAK,GAAK,KAAM,KAAX,IAAf,C;O;KAFrD,C;0FAGA,yB;MA+PA,kB  
AS6D,sB;MAT7D,+B;MA/PA,wB;QAEuD,OiBAA,eAAW,CjBsQ7B,eAAW,oBAAL,SAAK,CAAL,iBAAN,CiBt  
Q6B,MAAK,KjBAI,KiBAO,KAAX,CAAhB,C;O;KjBFvD,C;4FAIA,yB;MD6NA,6B;MC7NA,wB;QAEuD,OAM  
D,cAAK,IAAK,KAAK,GAAy,CD8N9C,cCpO+B,KDoOrB,KAAL,GAAiB,GAAtB,CC9N8C,MAAZ,IAAf,C;O;K  
ARtD,C;4FAGA,yB;MC2NA,6B;MD3NA,wB;QAEwD,OAGF,cAAK,IAAK,KAAK,GAAy,CC+N9C,cDIogC,KC  
kOtB,KAAL,GAAiB,KAAtB,CD/N8C,MAAZ,IAAf,C;O;KALtD,C;4FAGA,yB;MAAA,6B;MAAA,wB;QAEsD,q  
BAAK,IAAK,KAAK,GAAM,KAAM,KAAX,IAAf,C;O;KAFtD,C;4FAGA,yB;MAkPA,kBAS6D,sB;MAT7D,+B;  
MAIPA,wB;QAEwD,OiBAA,eAAW,CjByP9B,eAAW,oBAAL,SAAK,CAAL,iBAAN,CiBzP8B,MAAK,UjBAK,K  
iBAO,KAAX,CAAhB,C;O;KjBFxD,C;4FAIA,yB;MDgNA,6B;MChNA,wB;QAEuD,OAMD,cAAe,YAAV,IAAK,  
KAAK,EAAY,CDiN9C,cCvN+B,KDuNrB,KAAL,GAAiB,GAAtB,CCjN8C,MAAZ,CAAf,C;O;KARtD,C;4FAGA  
,yB;MC8MA,6B;MD9MA,wB;QAEwD,OAGF,cAAe,YAAV,IAAK,KAAK,EAAY,CCkN9C,cDrNgC,KCqNtB,K  
AAL,GAAiB,KAAtB,CDiN8C,MAAZ,CAAf,C;O;KALtD,C;4FAGA,yB;MAAA,6B;MAAA,wB;QAEsD,qBA Ae,  
YAAV,IAAK,KAAK,EAAM,KAAM,KAAX,CAAf,C;O;KAFtD,C;4FAGA,yB;MAqOA,kBAS6D,sB;MAT7D,+B;  
MARoA,wB;QAEwD,OiBAA,eAAW,CjB4O9B,eAAW,oBAAL,SAAK,CAAL,iBAAN,CiB5O8B,MAAK,UjBAK,  
KiBAO,KAAX,CAAhB,C;O;KjBFxD,C;wFAIA,yB;MDmMA,6B;MC7LA,4C;MANA,wB;QAEqD,OAMD,WAA  
W,IAAX,EDoMjB,cC1M2B,KD0MjB,KAAL,GAAiB,GAAtB,CCpMiB,C;O;KARpD,C;wFAGA,yB;MCiMA,6B;  
MD9LA,4C;MAHA,wB;QAEsD,OAGF,WAAW,IAAX,ECqMjB,cDxM4B,KCwMIB,KAAL,GAAiB,KAAtB,CDr  
MiB,C;O;KALpD,C;wFAGA,yB;MAAA,4C;MAAA,wB;QAEoD,kBAAW,IAAX,EAAiB,KAAjB,C;O;KAFpD,C;  
wFAGA,yB;MAwNA,kBAS6D,sB;MAT7D,+B;MiBxNA,8C;MjBAA,wB;QAEsD,OiBAA,YjB+NjB,eAAW,oBA  
AL,SAAK,CAAL,iBAAN,CiB/NiB,EjBAmb,KiBAnB,C;O;KjBFtD,C;wFAIA,yB;MDsLA,6B;MCxKA,kD;MadA  
,wB;QAMqD,OAcD,cAAc,IAAd,ED2KjB,cCzL2B,KDyLjB,KAAL,GAAiB,GAAtB,CC3KiB,C;O;KApBpD,C;wF  
AOA,yB;MCgLA,6B;MDzKA,kD;MAPA,wB;QAMsD,OAO,cAAc,IAAd,EC4KjB,cDnL4B,KCmLiB,KAAL,GA  
AiB,KAAtB,CD5KiB,C;O;KAbpD,C;wFAOA,yB;MAAA,kD;MAAA,wB;QAMoD,qBAAc,IAAd,EAAoB,KAApB  
,C;O;KANpD,C;wFAOA,yB;MA+LA,kBAS6D,sB;MAT7D,+B;MiB/LA,oD;MjBAA,wB;QAMsD,OiBAA,ejBkMj  
B,eAAW,oBAAL,SAAK,CAAL,iBAAN,CiBiMiB,EjBAmb,KiBAnB,C;O;KjBNtD,C;kGAQA,yB;MDyJA,6B;MC  
7LA,4C;MAoCA,wB;QAMiD,OAxCG,WAAW,IAAX,EDoMjB,cC5J4B,KD4JIB,KAAL,GAAiB,GAAtB,CCpMiB  
,C;O;KAKpD,C;kGAOA,yB;MCmJA,6B;MD9LA,4C;MA2CA,wB;QAMkD,OA/CE,WAAW,IAAX,ECqMjB,cDt  
J6B,KCsJnB,KAAL,GAAiB,KAAtB,CDrMiB,C;O;KAyCpD,C;kGAOA,yB;MAIDA,4C;MAkDA,wB;QAMgD,OA  
tDI,WAAW,IAAX,EAsDA,KAtDA,C;O;KAgDpD,C;kGAOA,yB;MAkKA,kBAS6D,sB;MAT7D,+B;MiBxNA,8C;  
MjBsDA,wB;QAMkD,OiB1DI,YjB+NjB,eAAW,oBAAL,SAAK,CAAL,iBAAN,CiB/NiB,EjB0DoB,KiB1DpB,C;  
O;KjBoDtD,C;wFAQA,yB;MD4HA,6B;MCxKA,kD;MDuOJ,0B;MAAA,+B;MC3LI,wB;QAQ6C,OD8LR,eAAW,  
OC5OI,cAAc,IAAd,ED2KjB,cC7HmB,KD6HT,KAAL,GAAiB,GAAtB,CC3KiB,CakLf,KD0DW,CAAX,C;O;KCt

MrC,C;wFASA,yB;MCoHA,6B;MDzKA,kD;MCwOJ,4B;MAAA,iC;MDnLI,wB;QAQ+C,OCsLR,gBAAY,QD7O  
C,cAAc,IAAd,EC4KjB,cDrHqB,KCqHX,KAAL,GAAiB,KAAtB,CD5KiB,CA4Lb,KCiDY,CAAZ,C;O;KD9LvC,C  
;wFASA,yB;MA9DA,kD;MA8DA,wB;QAQ2C,OAhES,cAAc,IAAd,EAgEL,KAhEK,C;O;KAwDpD,C;wFASA,y  
B;MA+HA,kBAS6D,sB;MAT7D,+B;MiB/LA,oD;MjBgEA,wB;QAQ6C,OiBIES,ejBkMjB,eAAW,oBAAL,SAAK,  
CAAL,iBAAN,CiBIMiB,EjBkEU,KiBIEV,C;O;KjB0DtD,C;wEAUA,yB;MAAA,6B;MAAA,mB;QAMyC,qBAAK  
,SAAK,QAAV,C;O;KANzC,C;wEAQA,yB;MAAA,6B;MAAA,mB;QAMyC,qBAAK,SAAK,QAAV,C;O;KANzC,  
C;gGAQA,yB;MAAA,8C;MAAA,wB;QAE6D,0BAAU,IAAV,EAAGB,KAhB,C;O;KAF7D,C;wFAIA,yB;MAA  
A,6B;MAAA,2B;QAOMD,qBAAK,aAAS,QAAc,C;O;KAPnD,C;wFASA,yB;MAAA,6B;MAAA,2B;QAOMD,qB  
AAK,cAAU,QAAf,C;O;KAPnD,C;wFASA,yB;MAAA,6B;MAAA,wB;QAEiD,qBAAK,IAAK,KAAL,GAAc,KAA  
M,KAAzB,C;O;KAFjD,C;sFAGA,yB;MAAA,6B;MAAA,wB;QAEgD,qBAAK,IAAK,KAAL,GAAa,KAAM,KAA  
xB,C;O;KAFhD,C;wFAGA,yB;MAAA,6B;MAAA,wB;QAEiD,qBAAK,IAAK,KAAL,GAAc,KAAM,KAAzB,C;O  
;KAFjD,C;wEAGA,yB;MAAA,6B;MAAA,mB;QAEgC,qBAAU,CAAL,SAAL,C;O;KAFhC,C;8EAlA,yB;MAAA,  
0B;MAAA,mB;QAUMC,OAAK,OAAL,SAAK,C;O;KAVxC,C;gFAWA,yB;MAAA,4B;MAAA,mB;QAUC,QAA  
K,QAAL,SAAK,C;O;KAVIC,C;4EAWA,Y;MASiC,gB;K;8EAcjC,yB;MAAA,kBASqD,sB;MATrD,mB;QASmC,  
OAAK,oBAAL,SAAK,CAAL,iB;O;KATnD,C;gFAWA,yB;MDwDJ,0B;MAAA,+B;MCxDI,mB;QASqC,OD0DA,  
eAAW,OC1DX,SD0DW,CAAX,C;O;KcErC,C;kFAUA,yB;MC+CJ,4B;MAAA,iC;MD/CI,mB;QASuC,OCiDA,g  
BAAY,QDjDZ,SCiDY,CAAZ,C;O;KD1DvC,C;8EAUA,Y;MAEmC,W;K;gFACnC,yB;MAAA,kBAS6D,sB;MAT7  
D,+B;MAAA,mB;QASqC,sBAAW,oBAAL,SAAK,CAAL,iBAAN,C;O;KATrC,C;gFAWA,yB;MASA,gD;MATA,  
mB;QAQqC,OAoe,aAAa,SAAb,C;O;KAFvC,C;kFASA,yB;MAAA,gD;MAAA,mB;QAMuC,oBAaA,SAAb,C;O;  
KANvC,C;8BAQA,Y;MAAYC,OArDD,oBAAL,SAAK,CAAL,iBAqDe,W;K;8BAhWtD,Y;MAAA,c;MAGqG,q  
D;MAHrG,a;K;4BAAA,iB;MAAA,2IAGqG,oCAHrG,G;K;SEAoWA,yB;MAAA,6B;MAAA,4B;QAWwC,qBAAU  
,SAAV,C;O;KAXxC,C;wEAYA,yB;MAAA,6B;MAAA,4B;QAWyC,qBAAU,SAAV,C;O;KAXzC,C;wEAYA,yB;  
MAAA,6B;MAAA,4B;QAUC,qBAAK,SAAL,C;O;KAVvC,C;wEAWA,yB;MAAA,6B;MAAA,4B;QAWwC,qB  
AAK,SAAK,QAAV,C;O;KAXxC,C;uEaA,yB;MAAA,gD;MAAA,4B;QASyC,oBAaKB,SAAlB,C;O;KATzC,C;  
wEAUA,yB;MAAA,gD;MAAA,4B;QAS0C,oBAaA,SAAb,C;O;KAT1C,C;liC3ZA,4B;MACqB,sB;K;sCAKjB,iB;  
MAM4C,OjCuXT,SiCvXS,aAAQ,KAAR,CjCuXT,C;K;sCiCrXnC,wB;MAOI,aAAQ,KAAR,IAAiB,KjCyQY,K;K;  
iFiCrQH,Y;MAAQ,OAAA,YAAQ,O;K;mCAE9C,Y;MAC6E,8BAAS,YAAT,C;K;IAEvD,mC;MAAC,oB;MACnB  
,eAAoB,C;K;2CACpB,Y;MAAYB,sBAAQ,YAAM,O;K;wCACvC,Y;MAAoD,Q;MAA9B,IAAI,eAAQ,YAAM,OA  
AlB,C;QAAA,OjCkWS,SiCIWe,aAAM,mBAAN,EAAM,2BAAN,OjCkWf,C;;QiCIW4C,MAAM,2BAAuB,YAAM  
,WAA7B,C;K;;yCAGrF,mB;MAIS,Q;MAAL,IAAI,eAAC,0EAAD,OAAJ,C;QAAGC,OAAO,K;MAEvC,OAAe,W  
AAR,YAAQ,EAAS,OjCoPK,KiCpPd,C;K;8CAGnB,oB;MACY,Q;MAA2B,gBAA3B,gE;MAA2B,c;;Qf6nDvB,U;  
QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,aAAO,I;UAAp,e;;QACrB,6B;QAaHb,OAAGB,gBAaHb,C;UAAgB  
,2B;Ue7nD6B,2Bf6nDR,Oe7nDQ,O;UAAA,W;YAAsB,oBAAR,YAAQ,Ef6nD9B,OIB74CJ,KiChPkC,C;;Uf6nD7  
C,IAAI,OAAJ,C;YAAyB,aAAO,K;YAAP,e;;QAC/C,aAAO,I;;Me9nDH,iB;K;kCAGJ,Y;MAAKC,OAAA,IAAK,  
QAAQ,OAAb,KAAqB,C;K;;IA9CvD,qC;MAAA,mD;MACgC,sBAAK,eAAS,IAAT,CAAL,C;MADhC,Y;K;;mC  
APJ,Y;MAAA,OAKqB,oDALrB,M;K;mCAAA,Y;MAAA,c;MAKqB,wD;MALrB,a;K;iCAAA,iB;MAAA,2IAKqB  
,0CALrB,G;K;8EAwDA,yB;MAAA,uC;MAWOC,wC;QAAA,wB;UAAW,OAAA,aAAK,KAAL,CjC+NV,K;S;O;  
MiC1OrC,6B;QAWI,OAAO,mBAAU,gCAAS,IAAT,GAAe,sBAaf,CAA V,C;O;KAXX,C;gFACa,oB;MAGkE,e;K  
;I+LjE5C,wC;MA8BIB,iC;MA9BsD,2BAAGB,KAaHb,EAaUB,YAAvB,EAaQC,CAArC,C;K;kFAC7B,Y;MAAQ,  
iB;K;yFACD,Y;MAAQ,gB;K;yFAKR,Y;MACxB,Q;MAAJ,IAAI,yCAAQ,4BAAK,UAAb,QAAJ,C;QpNmHyC,M  
AAM,6BoNnHb,6EpNmH2C,WAA9B,C;;MoNIH/C,OhOoDiD,SgOpD1C,ShOoDoD,KAAK,GAAW,CgOpD7D,  
WhOoD6D,MAAX,IAAf,C;K;2CgOjDrD,iB;MAA8C,WhO+BoB,YgO/BpB,UhO+BqC,KAAjB,EgO/BX,KhO+B  
wC,KAA7B,CgO/BpB,K;MAAA,S;QAaKB,OhO+BE,YgO/BF,KhO+BmB,KAAjB,EgO/BO,ShO+Bsb,KAA7B,C  
gO/BF,K;;MAAlB,W;K;kCAE9C,Y;MAKkC,OhOwBgC,YgOxBhC,UhOwBiD,KAAjB,EgOxBxB,ShOwBqD,KA  
A7B,CgOxBhC,I;K;iCAEIC,iB;MAEY,UAAwB,M;MADhC,2CAAuB,kBAaA,KAAM,UAA nB,KACf,2CAAS,KA  
AM,MAAf,cAAwB,6CAAQ,KAAM,KAA d,QAAxB,CADe,CAA vB,C;K;mCAGJ,Y;MACI,OAAI,cAAJ,GAAe,E  
AAf,GAAwB,MAAK,UhOgQA,KgOhQL,QAAqB,ShOgQhB,KgOhQL,I;K;mCAE5B,Y;MAAKC,OAAE,UAAf,q  
BAAU,S;K;IAE5C,+B;MAAA,mC;MACI,aAC8B,cAAU,4BAAK,UAAf,EAA0B,4BAAK,UAA/B,C;K;;IAFIC,2C

;MAAA,0C;QAAA,yB;;MAAA,mC;K;;IAYJ,oD;MA4CI,uC;MAiCI,IAAI,SAAQ,CAAZ,C;QAAuB,MAAA,gCAA  
yB,wBAAZB,C;MACpC,IAAI,SAAQ,WAAZ,C;QAA2B,MAAA,gCAAYB,wEAAzB,C;MAG5C,aAGyB,K;MAEz  
B,YAGwB,4BAA0B,KAA1B,EAAiC,YAAjC,EAA+C,IAA/C,C;MAExB,YAGuB,I;K;yCAEvB,Y;MAAgD,mCA  
AwB,UAAxB,EAA+B,SAA/B,EAAqC,SAArC,C;K;wCAEhD,Y;MAMqC,OAAI,YAAO,CAAX,GhOhC6B,YgOg  
Cf,UhOhCgC,KAAjB,EgOgCP,ShOhCoC,KAA7B,CgOgCf,IAAd,GhOhC6B,YgOgCG,UhOhCc,KAAjB,EgOgCW  
,ShOhCkC,KAA7B,CgOgCG,I;K;uCAErE,iB;MAEY,UAAwB,M;MADhC,iDAA6B,kBAaA,KAAM,UAAAnB,KA  
CrB,2CAAS,KAAM,MAAf,cAAwB,6CAAQ,KAAM,KAAd,QAAxB,KAA8C,cAAQ,KAAM,KADvC,CAA7B,C;  
K;yCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,OAAM,MAAK,UhOwMN,KgOxMC,QAAqB,ShOwMtB,  
KgOxMC,IAAN,SAAgD,SAAhD,I;K;yCAE5B,Y;MAAkC,OAAI,YAAO,CAAX,GAAgB,UAAF,qBAAU,SAAV,c  
AAqB,SAAnc,GAAgD,UAAF,2BAAgB,SAAhB,eAA4B,CAAC,SAAD,IAA5B,C;K;IAEHf,qC;MAAA,yC;K;kEA  
CI,sC;MAQ2F,2BAAgB,UAAhB,EAA4B,QAA5B,EAA5C,IAAtC,C;K;;;IAT/F,iD;MAAA,gD;QAAA,+B;;MAAA,  
yC;K;;IAmBiC,oD;MACjC,sBAA2B,I;MAC3B,iBAAmC,OAAO,CAA1C,GhOhEkE,YgOgErB,KhOhEsC,KAAjB  
,EgOgEZ,IhOhEyC,KAA7B,CgOgErB,KAA7C,GhOhEkE,YgOgEF,KhOhEmB,KAAjB,EgOgEO,IhOhEsB,KAA7  
B,CgOgEF,K;MACHe,chOmRmC,SgOnRhB,IhOmRgB,C;MgOlRnC,cAAuB,cAAJ,GAAa,KAAb,GAAwB,mB;K;  
gDAE3C,Y;MAAkC,qB;K;6CAEIC,Y;MACI,YAA Y,W;MACZ,IAAI,6BAAS,mBAAT,QA AJ,C;QACI,IAAI,CAA  
C,cAAL,C;UAAc,MAAA,6B;QAC3B,iBAAU,K;;QAEV,chO1D6C,SgO0D7C,WhO1DuD,KAAK,GgO0DpD,WhO  
1D+D,KAA X,IAAf,C;;MgO4DjD,OAAO,K;K;;I/M7HU,qB;MAErB,6B;MAFkG,gB;K;IAEIG,2B;MAAA,+B;MA  
CI,iBAGoC,a;MAEpC,iBAGoC,c;MAEpC,kBAGmC,C;MAEnC,iBAGkC,E;K;;IANtC,uC;MAAA,sC;QAAA,qB  
;;MAAA,+B;K;sGAsBA,yB;MIBqRA,WAS6D,wB;MAT7D,+B;MkB7PA,gD;MAxBa,wB;QAM0D,OAYBS,aAAa  
,IAAK,KAAIB,EAA8B,CIB+P5D,eAAW,oBkBxRyB,KIBwR9B,KAAK,CAAL,UAAAN,CkCk/P4D,MAA9B,C;O;K  
A/BnE,C;sGAQA,yB;MhB8QA,aAS6D,0B;MAT7D,+B;MgB9PA,gD;MAhBA,wB;QAM2D,OAIbQ,aAAa,IAAK,  
KAAIB,EAA8B,ChBgQ5D,eAAW,oBgBjR0B,KhBiR/B,KAAK,CAAL,YAAN,CgBhQ4D,MAA9B,C;O;KAvBnE,  
C;sGAQA,yB;MjByRA,kBAS6D,sB;MAT7D,+B;MiBjRA,gD;MARA,wB;QAMyD,OASU,aAAa,IAAK,KAAIB,E  
AA8B,CjBmR5D,eAAW,oBiB5RwB,KjB4R7B,KAAK,CAAL,iBAAN,CiBnR4D,MAA9B,C;O;KAFnE,C;kGAQA,  
yB;MAAA,gD;MAAA,wB;QAOMe,oBAAa,IAAK,KAAIB,EAAwB,KAAM,KAA9B,C;O;KAPnE,C;4FASA,yB;  
MIBoPA,WAS6D,wB;MAT7D,+B;MkBPpA,wB;QAEuD,OASA,eAAM,IAAK,KAAK,KAAW,CIBkP7C,eAAW,o  
BkB3PiB,KIB2PiB,KAAK,CAAL,UAAAN,CkBiP6C,MAAX,CAAhB,C;O;KAXvD,C;4FAGA,yB;MhBkPA,aAS6D  
,0B;MAT7D,+B;MgBIPA,wB;QAEwD,OAMD,eAAM,IAAK,KAAK,KAAW,ChBmP7C,eAAW,oBgBzPkB,KhBy  
PvB,KAAK,CAAL,YAAN,CgBnP6C,MAAX,CAAhB,C;O;KARvD,C;4FAGA,yB;MjBkQA,kBAS6D,sB;MAT7D,  
+B;MiBIQA,wB;QAEsD,OAGC,eAAM,IAAK,KAAK,KAAW,CjBsQ7C,eAAW,oBiBzQgB,KjByQrB,KAAK,CA  
AL,iBAAN,CiBtQ6C,MAAX,CAAhB,C;O;KALvD,C;4FAGA,yB;MAAA,+B;MAAA,wB;QAEuD,sBAAM,IAAK  
,KAAK,KAAK,KAAM,KAAX,CAAhB,C;O;KAFvD,C;8FAIA,yB;MIBuOA,WAS6D,wB;MAT7D,+B;MkBVoa,  
wB;QAEwD,OASA,eAAM,IAAK,KAAK,UAA Y,CIBqO/C,eAAW,oBkB9OmB,KIB8OxB,KAAK,CAAL,UAAAN,  
CkBrO+C,MAAZ,CAAhB,C;O;KAXxD,C;8FAGA,yB;MhBqOA,aAS6D,0B;MAT7D,+B;MgBrOA,wB;QAEyD,O  
AMD,eAAM,IAAK,KAAK,UAA Y,ChBsO/C,eAAW,oBgB5OoB,KhB4OzB,KAAK,CAAL,YAAN,CgBtO+C,MA  
AZ,CAAhB,C;O;KARxD,C;8FAGA,yB;MjBqPA,kBAS6D,sB;MAT7D,+B;MiBrPA,wB;QAEuD,OAGC,eAAM,I  
AAK,KAAK,UAA Y,CjByP/C,eAAW,oBiB5PkB,KjB4PvB,KAAK,CAAL,iBAAN,CiBzP+C,MAAZ,CAAhB,C;O;  
KALxD,C;8FAGA,yB;MAAA,+B;MAAA,wB;QAEwD,sBAAM,IAAK,KAAK,UAA M,KAAM,KAAZ,CAAhB,C;  
O;KAFxD,C;8FAIA,yB;MIB0NA,WAS6D,wB;MAT7D,+B;MkB1NA,wB;QAEwD,OASA,eAAM,IAAK,KAAK,  
UAA Y,CIBwN/C,eAAW,oBkBJOmB,KIBiOxB,KAAK,CAAL,UAAAN,CkBXN+C,MAAZ,CAAhB,C;O;KAXxD,C;  
8FAGA,yB;MhBwNA,aAS6D,0B;MAT7D,+B;MgBxNA,wB;QAEyD,OAMD,eAAM,IAAK,KAAK,UAA Y,ChBy  
N/C,eAAW,oBgB/NoB,KhB+NzB,KAAK,CAAL,YAAN,CgBzN+C,MAAZ,CAAhB,C;O;KARxD,C;8FAGA,yB;  
MjBwOA,kBAS6D,sB;MAT7D,+B;MiBxOA,wB;QAEuD,OAGC,eAAM,IAAK,KAAK,UAA Y,CjB4O/C,eAAW,o  
BiB/OkB,KjB+OvB,KAAK,CAAL,iBAAN,CiB5O+C,MAAZ,CAAhB,C;O;KALxD,C;8FAGA,yB;MAAA,+B;MA  
AA,wB;QAEwD,sBAAM,IAAK,KAAK,UAA M,KAAM,KAAZ,CAAhB,C;O;KAFxD,C;0FAIA,yB;MIB6MA,WA  
S6D,wB;MAT7D,+B;MkBPpMA,8C;MATA,wB;QAEsD,OASA,YAA Y,IAAZ,EIB2MjB,eAAW,oBkBPpNe,KIBoNp  
B,KAAK,CAAL,UAAAN,CkB3MiB,C;O;KAXtD,C;0FAGA,yB;MhB2MA,aAS6D,0B;MAT7D,+B;MgBrMA,8C;M  
ANA,wB;QAEuD,OAMD,YAA Y,IAAZ,EhB4MjB,eAAW,oBgBINgB,KhBkNrB,KAAK,CAAL,YAAN,CgB5MiB

,C;O;KARtD,C;0FAGA,yB;MjB2NA,kBAS6D,sB;MAT7D,+B;MiBxNA,8C;MAHA,wB;QAEqD,OAGC,YAAAY,I  
AAZ,EjB+NjB,eAAW,oBiBIOc,KjBkOnB,KAACK,CAAL,iBAAN,CiB/NiB,C;O;KALtD,C;0FAGA,yB;MAAA,8C;  
MAAA,wB;QAEsD,mBAAY,IAAZ,EAakB,KAAIB,C;O;KAFtD,C;0FAIA,yB;MIBgMA,WAS6D,wB;MAT7D,+  
B;MkB3KA,oD;MArBA,wB;QAMsD,OAqBA,eAAe,IAAf,EIB8KjB,eAAW,oBkbnMe,KIBmMpB,KAACK,CAAL,  
UAAN,CkB9KiB,C;O;KA3BtD,C;0FAOA,yB;MhB0LA,aAS6D,0B;MAT7D,+B;MgB5KA,oD;MAdA,wB;QAMu  
D,OAcD,eAAe,IAAf,EhB+KjB,eAAW,oBgB7LgB,KhB6LrB,KAACK,CAAL,YAAN,CgB/KiB,C;O;KApBtD,C;0F  
AOA,yB;MjBsMA,kBAS6D,sB;MAT7D,+B;MiB/LA,oD;MAPA,wB;QAMqD,OAOC,eAAe,IAAf,EjBkMjB,eAA  
W,oBiBzMc,KjByMnB,KAACK,CAAL,iBAAN,CiBIMiB,C;O;KAbtD,C;0FAOA,yB;MAAA,oD;MAAA,wB;QAMs  
D,sBA Ae,IAAf,EAAqB,KAArB,C;O;KANtD,C;oGAQA,yB;MIBmKA,WAS6D,wB;MAT7D,+B;MkBpMA,8C;M  
AiCA,wB;QAMkD,OArCI,YAAAY,IAAZ,EIB2MjB,eAAW,oBkbtKgB,KIBsKrB,KAACK,CAAL,UAAN,CkB3MiB,  
C;O;KA+BtD,C;oGAOA,yB;MhB6JA,aAS6D,0B;MAT7D,+B;MgBrMA,8C;MAwCA,wB;QAMmD,OA5CG,YA  
AY,IAAZ,EhB4MjB,eAAW,oBgBhKiB,KhBgKtB,KAACK,CAAL,YAAN,CgB5MiB,C;O;KAsCtD,C;oGAOA,yB;  
MjByKA,kBAS6D,sB;MAT7D,+B;MiBxNA,8C;MA+CA,wB;QAMiD,OAnDK,YAAAY,IAAZ,EjB+NjB,eAAW,oB  
iB5Ke,KjB4KpB,KAACK,CAAL,iBAAN,CiB/NiB,C;O;KA6CtD,C;oGAOA,yB;MatDA,8C;MAsDA,wB;QAMkD,  
OA1DI,YAAAY,IAAZ,EA0DA,KA1DA,C;O;KAoDtD,C;0FAQA,yB;MIBsIA,WAS6D,wB;MAT7D,+B;MkB3KA,o  
D;MIB4OJ,0B;MAAA,+B;MkBVMI,wB;QAQ6C,OIB0MP,eAAW,OkBjPK,eAAe,IAAf,EIB8KjB,eAAW,oBkBV  
I  
M,KIBuIX,KAACK,CAAL,UAAN,CkB9KiB,CA4KjB,KIBqEY,SAAX,C;O;KkBINtC,C;0FASA,yB;MhB8HA,aAS  
6D,0B;MAT7D,+B;MgB5KA,oD;MhB6OJ,4B;MAAA,iC;MgB/LI,wB;QAQ+C,OhBkMP,gBAAY,QgBIPE,eAAe,I  
AAf,EhB+KjB,eAAW,oBgB/HQ,KhB+Hb,KAACK,CAAL,YAAN,CgB/KiB,CAsLf,KhB4Da,SAAZ,C;O;KGB1Mx  
C,C;0FASA,yB;MjBwIA,kBAS6D,sB;MAT7D,+B;MiB/LA,oD;MjBkQJ,6B;MiB3MI,wB;QAQ2C,OjB8MP,ciBvQ  
kB,eAAe,IAAf,EjBkMjB,eAAW,oBiBzII,KjByIT,KAACK,CAAL,iBAAN,CiBIMiB,CagMnB,KjBuEW,QAAV,C;  
O;KiBtNpC,C;0FASA,yB;MAhEA,oD;MAGEA,wB;QAQ6C,OAIES,eAAe,IAAf,EAkEL,KAIEK,C;O;KA0DtD,C;  
0EAUA,yB;MAAA,+B;MAAA,mB;QAM0C,sBAAM,SAAK,MAAX,C;O;KAN1C,C;0EAQA,yB;MAAA,+B;MA  
AA,mB;QAM0C,sBAAM,SAAK,MAAX,C;O;KAN1C,C;kGAQA,yB;MAAA,gD;MAAA,wB;QAE+D,2BAAW,I  
AAX,EAAiB,KAAjB,C;O;KAF/D,C;0FAIA,yB;MAAA,+B;MAAA,2B;QAOoD,sBAAM,oBAAS,QAAT,CAAN,C  
;O;KAPpD,C;0FASA,yB;MAAA,+B;MAAA,2B;QAOoD,sBAAM,6BAAU,QAAV,CAAN,C;O;KAPpD,C;0FASA,  
yB;MAAA,+B;MAAA,wB;QAEmD,sBAAM,IAAK,KAAL,KAAC,KAAM,KAAPB,CAAN,C;O;KAFnD,C;wFAG  
A,yB;MAAA,+B;MAAA,wB;QAEkD,sBAAM,IAAK,KAAL,IAAa,KAAM,KAAnB,CAAN,C;O;KAFID,C;0FAG  
A,yB;MAAA,+B;MAAA,wB;QAEmD,sBAAM,IAAK,KAAL,KAAC,KAAM,KAAPB,CAAN,C;O;KAFnD,C;0EA  
GA,yB;MAAA,+B;MAAA,mB;QAEiC,sBAAM,SAAK,MAAX,C;O;KAFjC,C;gFAIA,yB;MAAA,0B;MAAA,mB;  
QAUmC,OAAK,OAAL,SAAK,S;O;KAVx C,C;kFAWA,yB;MAAA,4B;MAAA,mB;QAUqC,OAAK,QAAL,SAAK  
,S;O;KAV1C,C;8EAWA,Y;MAUic,OAAA,SAAK,Q;K;gFACtC,Y;MASmC,gB;K;kFAEnC,yB;MIBmEJ,0B;MA  
AA,+B;MkbnEI,mB;QASqC,OIBqEC,eAAW,OkBrEZ,SIBqEY,SAAX,C;O;KkB9EtC,C;0FAUA,yB;MhB0DJ,4B;  
MAAA,iC;MgB1DI,mB;QASuC,OhB4DC,gBAAY,QgB5Db,ShB4Da,SAAZ,C;O;KGBrExC,C;gFAUA,yB;MjBqE  
J,6B;MiBrEI,mB;QASmC,OjBuEC,ciBvED,SjBuEW,QAAV,C;O;KiBhFpC,C;kFAUA,Y;MAEQC,W;K;kFAErC,y  
B;MASA,kD;MATA,mB;QAQqC,OASE,cAAc,SAAd,C;O;KAjBvC,C;0FASA,yB;MAAA,kD;MAAA,mB;QAQu  
C,qBAAC,SAAd,C;O;KARvC,C;+BAUA,Y;MAAyC,qBAAC,SAAd,C;K;::;+BAnW7C,Y;MAAA,c;MAGsG,qD;  
MAHtG,a;K;6BAAA,iB;MAAA,2IAGsG,oCAHtG,G;K;wEAuWA,yB;MAAA,+B;MAAA,4B;QAW0C,sBAAW,o  
BAAL,SAAK,CAAX,C;O;KAX1C,C;0EAYA,yB;MAAA,+B;MAAA,4B;QAW2C,sBAAW,oBAAL,SAAK,CAA  
X,C;O;KAX3C,C;0EAYA,yB;MAAA,+B;MAAA,4B;QAWyC,sBAAW,oBAAL,SAAK,CAAX,C;O;KAXzC,C;0E  
AYA,yB;MAAA,+B;MAAA,4B;QAU0C,sBAAM,SAAN,C;O;KAV1C,C;yEAYA,yB;MAAA,kD;MAAA,4B;QAS  
2C,qBAAmB,SAAnB,C;O;KAT3C,C;0EAUA,yB;MAAA,kD;MAAA,4B;QAS4C,qBAAC,SAAd,C;O;KAT5C,C;Ii  
B9ZA,6B;MACqB,sB;K;uCAKjB,iB;MAM6C,OjBsYP,UiBtYO,aAAQ,KAAR,CjBsYP,C;K;uCiBpYtC,wB;MAOI  
,aAAQ,KAAR,IAAiB,KjBoRc,K;K;kFiBhRL,Y;MAAQ,OAAA,YAAQ,O;K;oCAE9C,Y;MAC8E,+BAAS,YAAT,  
C;K;IAExD,oC;MAAC,oB;MACnB,eAAoB,C;K;4CACpB,Y;MAAyB,sBAAQ,YAAM,O;K;yCACvC,Y;MAAoD,  
Q;MAA9B,IAAI,eAAQ,YAAM,OAAIB,C;QAAA,OjBiXY,UiBjXY,aAAM,mBAAN,EAAM,2BAAN,OjBiXZ,C;;  
QibjX0C,MAAM,2BAAUb,YAAM,WAA7B,C;K;;0CAGtF,mB;MAIS,Q;MAAL,IAAI,eAAC,0EAAD,QAAJ,C;Q  
AAiC,OAAO,K;MAExC,OAAe,WAAR,YAAQ,EAAS,OjB+PO,KiB/PhB,C;K;+CAGnB,oB;MACY,Q;MAA2B,g

BAA3B,gE;MAA2B,c;;QhB6nDvB,U;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,6B ;QAAhB,OAAgB,gBAAhB,C;UAAgB,2B;UgB7nD6B,2BhB6nDR,OgB7nDQ,Q;UAAA,W;YAAuB,oBAAR,YAA Q,EhB6nD/B,ODI4CF,KiB3PiC,C;;UhB6nD9C,IAAI,OAAJ,C;YAAyB,aAAO,K;YAAP,e;;QAC/C,aAAO,I;;MgB 9nDH,iB;K;mCAGJ,Y;MAAkC,OAAA,IAAK,QAAQ,OAAb,KAAqB,C;K;;IA9CvD,sC;MAAA,oD;MACgC,uBA AK,iBAAU,IAAV,CAAL,C;MADhC,Y;K;;;oCAPJ,Y;MAAA,OAKqB,qDALrB,M;K;oCAAA,Y;MAAA,c;MAKq B,wD;MALrB,a;K;kCAAA,iB;MAAA,2IAKqB,0CALrB,G;K;gFAwDA,yB;MAAA,yC;MAWsC,yC;QAAA,wB;U AAW,OAAA,aAAK,KAAL,CjB00V,K;S;O;MiBrPvC,6B;QAWI,OAAO,oBAAW,kBAAU,IAAV,EAAgB,uBAAh B,CAAX,C;O;KAXX,C;kFAcA,oB;MAGqE,e;K;I-LjE9C,2C;MA8BnB,kC;MA9ByD,4BAAiB,KAAjB,EAAwB, YAAxB,K;K;qFAC/B,Y;MAAQ,iB;K;4FACD,Y;MAAQ,gB;K;4FAKR,Y;MACzB,Q;MAAJ,IAAI,yCAAQ,6BAA M,UAAAd,QAAJ,C;QrNmHyC,MAAM,6BqNnHZ,6ErNmH0C,WAA9B,C;;MqNIH/C,OhNuDmD,UgNvD5C,ShNu DuD,KAAK,KAAW,CjBsQ7C,UAAW,oBAAL,CiO7TzB,WjO6TyB,MAAK,CAAL,iBAAN,CiBtQ6C,MAAX,CA AhB,C;K;8CgNpDvD,iB;MAA+C,WhNuCoB,agNvCpB,UhNuCsC,KAAIB,EgNvCX,KhNuCyC,KAA9B,CgNvCp B,K;MAAA,S;QAAkB,OhNuCE,agNvCF,KhNuCoB,KAAIB,EgNvCO,ShNuCuB,KAA9B,CgNvCF,K;;MAAIB,W ;K;qCAE/C,Y;MAKkC,OhNgCiC,agNhCjC,UhNgCmD,KAAIB,EgNhCzB,ShNgCuD,KAA9B,CgNhCjC,I;K;oCA EIC,iB;MAEY,UAAwB,M;MADhC,8CAAwB,kBAAa,KAAM,UAAAnB,KACHB,2CAAS,KAAM,MAAf,cAAwB,6 CAAQ,KAAM,KAAd,QAAxB,CADgB,CAAXB,C;K;sCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,MhNiQ K,CARckB,UgN5NjB,UhN4N4B,KAAL,KAAoB,CAVzB,UgNINP,UhNkNa,yBgNINH,EhNkNG,CAAN,CAUyB, MAAPB,CAAN,CAqCIB,MAAK,QgNjQV,QhNiQK,CARckB,UgN5NoB,ShN4NT,KAAL,KAAoB,CAVzB,UgNI N6B,ShNkNvB,yBgNINgC,EhNkNhC,CAAN,CAUyB,MAAPB,CAAN,CAqCIB,MAAK,QgNjQV,I;K;sCAE5B,Y; MAAkC,OAAE,UAAF,qBAAU,S;K;IAE5C,gC;MAAA,oC;MACI,aAC+B,iBAAW,6BAAM,UAAjB,EAA4B,6BA AM,UAAIC,C;K;;IAFnC,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;;IAYJ,qD;MA4CI,wC;MAiCI,IAAI,gBAAJ,C; QAAwB,MAAA,gCAAYB,wBAAzB,C;MACrC,IAAI,sCAAJ,C;QAA4B,MAAA,gCAAYB,yEAAzB,C;MAG7C,aA G0B,K;MAE1B,YAGyB,4BAA0B,KAA1B,EAAiC,YAAjC,EAA+C,IAA/C,C;MAEzB,YAGwB,I;K;0CAExB,Y; MAAiD,oCAAYB,UAAzB,EAAgC,SAAhC,EAAcC,SAATC,C;K;yCAEjD,Y;MAMqC,OAAI,uBAAO,CAAX,GhN xB8B,agNwBhB,UhNxBkC,KAAIB,EgNwBR,ShNxBsC,KAA9B,CgNwBhB,IAAd,GhNxB8B,agNwBE,UhNxBg B,KAAIB,EgNwBU,ShNxBoB,KAA9B,CgNwBE,I;K;wCAErE,iB;MAEY,UAAwB,M;MADhC,kDAA8B,kBAAa, KAAM,UAAAnB,KACTB,2CAAS,KAAM,MAAf,cAAwB,6CAAQ,KAAM,KAAd,QAAxB,KAA8C,kBAAQ,KAA M,KAAd,CADxB,CAA9B,C;K;0CAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,OAAM,MhNyMD,CARckB, UgNpKX,UhNoKsB,KAAL,KAAoB,CAVzB,UgN1JD,UhN0JO,yBgN1JG,EhN0JH,CAAN,CAUyB,MAAPB,CAA N,CAqCIB,MAAK,QgNzMJ,QhNyMD,CARckB,UgNpK0B,ShNoKf,KAAL,KAAoB,CAVzB,UgN1JmC,ShN0J7B ,yBgN1JsC,EhN0JtC,CAAN,CAUyB,MAAPB,CAAN,CAqCIB,MAAK,QgNzMJ,IAAN,SAAqF,cAAU,6BAAU,E AAV,CAAV,CAAYB,QAA9G,I;K;0CAE5B,Y;MAAkC,OAAI,uBAAO,CAAX,GAAgB,UAAF,qBAAU,SAAV,cA AqB,SAArB,WAAAd,GAAgD,UAAF,2BAAgB,SAAhB,cAA6B,SAAD,aAA5B,W;K;IAEHf,sC;MAAA,0C;K;mEA CI,sC;MAQ+F,4BAAiB,UAAjB,EAA6B,QAA7B,EAAuC,IAAvC,C;K;;IATnG,kD;MAAA,iD;QAAA,gC;;MAA A,0C;K;;IAmBkC,qD;MACIC,sBAA2B,I;MAC3B,iBAAMc,kBAAO,CAA1C,GhNxDMe,agNwDtB,KhNxDwC,K AAIB,EgNwDb,IhNxD2C,KAA9B,CgNwDtB,KAA7C,GhNxDMe,agNwDH,KhNxDqB,KAAIB,EgNwDM,IhNxD wB,KAA9B,CgNwDH,K;MACHe,chNkSsC,UgNISnB,IhNkSmB,C;MgNjStC,cAAuB,cAAJ,GAAa,KAAb,GAAw B,mB;K;iDAE3C,Y;MAAkC,qB;K;8CAEIC,Y;MACI,YAAY,W;MACZ,IAAI,6BAAS,mBAAT,QAAJ,C;QACI,IA AI,CAAC,cAAL,C;UAAc,MAAA,6B;QAC3B,iBAAU,K;;QAEV,chNvD+C,UgNuD/C,WhNvD0D,KAAK,KgNuD vD,WhNvDkE,KAAX,CAAhB,C;;MgNyDnD,OAAO,K;K;;wECrHf,yB;MAAA,8C;MAAA,uB;QAOI,OAAO,MAA M,CAAN,EAAS,CAAT,C;O;KAPX,C;wEAUA,yB;MAAA,8C;MAAA,uB;QAOI,OAAO,MAAM,CAAN,EAAS,C AAT,C;O;KAPX,C;wEAUA,yB;MAAA,8C;MAAA,uB;QAOI,OAAO,MAAM,CAAN,EAAS,CAAT,C;O;KAPX,C ;wEAUA,yB;MAAA,8C;MAAA,uB;QAOI,OAAO,MAAM,CAAN,EAAS,CAAT,C;O;KAPX,C;oFC7BA,yB;MAA A,gD;MAAA,4B;QAM6C,OAAQ,anO+RhB,cmO/RgB,C;O;KANrD,C;oGAQA,yB;MpHwCA,4B;MoHxCA,4B;Q AMqD,OpHwCM,Y/G+OtB,c+G/OsB,C;O;KoH9C3D,C;sGAQA,yB;MAAA,ke;MAAA,4B;QAMsD,OAAQ,sBnO +QzB,cmO/QyB,C;O;KAN9D,C;8FAQA,yB;MAAA,0D;MnOwWA,6B;MmOxWA,4B;QAOmD,OnO2WZ,cmO3 WoB,kBnOsQtB,cmOtQsB,CnO2WpB,C;O;KmOIXvC,C;4FASA,yB;MAAA,wD;MnO+VA,6B;MmO/VA,4B;QA OkD,OnOkWX,cmOIWmB,iBnO6PrB,cmO7PqB,CnOkWnB,C;O;KmOzWvC,C;gFASA,yB;MAAA,4C;MnOsVA,

6B;MmOtVA,sC;QAayD,OnOmV1B,cmOnV0B,WnO8O5B,cmO9O4B,EAAW,QAAX,CnOmV1B,C;O;KmOhWv  
C,C;kFAgBA,yB;MAAA,8C;MnOsUA,6B;MmOtUA,sC;QAa0D,OnOmUnB,cmOnU2B,YnO8N7B,cmO9N6B,EA  
AY,QAAX,CnOmU3B,C;O;KmOhVvC,C;oFAgBA,yB;MAAA,gD;MAAA,4B;QAM8C,OAAS,alNgOhB,ckNhOg  
B,C;O;KANvD,C;oGAQA,yB;MAAA,gE;MAAA,4B;QAMsD,OAAS,qBINwNxB,ckNxnwB,C;O;KAN/D,C;sGA  
QA,yB;MAAA,kE;MAAA,4B;QAMuD,OAAS,sBINgNzB,ckNHNyB,C;O;KANhE,C;8FAQA,yB;MAAA,0D;MIN  
6SA,+B;MkN7SA,4B;QAQd,OlNgTX,ekNhToB,kBINuMvB,ckNvMuB,CINgTpB,C;O;KkNvT1C,C;4FASA,yB;  
MAAA,wD;MINoSA,+B;MkNpSA,4B;QAoOd,OlNuSV,ekNvSmB,iBIN8LtB,ckN9LsB,CINuSnB,C;O;KkN9S1C,  
C;+EASA,yB;MAAA,4C;MIN2RA,+B;MkN3RA,sC;QAa2D,OlNwRjB,ekNxr0B,WIN+K7B,ckN/K6B,EAAW,Q  
AAX,CINwR1B,C;O;KkNrS1C,C;iFAeA,yB;MpHgEA,4C;M9F4MA,+B;MkN5QA,sC;QAa4D,OlNyQIB,e8FzMu  
B,W9FgG1B,c8FhG0B,EAAW,CoHhEK,QpHgEL,IAAX,C9FyMvB,C;O;KkNtR1C,C;oFAeA,yB;MpOwJI,6B;Mo  
O1SJ,gD;MAkJA,4B;QAM8C,OAIJO,anO+RhB,CDcE,cAAU,cAAL,GAAiB,GAAtB,CCdF,MmO/RgB,C;O;KA4I  
rD,C;oGAQA,yB;MpH1GA,4B;MoH0GA,4B;QAMsD,OpH1GK,YhHuMpB,c8N1Ge,GAAAY,G9G7FP,C8G6FN,G  
AA6C,EAA7C,I;O;KMOrD,C;sGAQA,yB;MNBa,kE;MMaA,4B;QAMuD,ONbkB,sB9NkGIC,c8NIGgB,GAAW,G  
AAO,C;O;KMOzE,C;8FAQA,yB;MAAA,0D;MpO+LA,0B;MAAA,+B;MoO/LA,4B;QAQd,OpOmMZ,eAAW,O  
oOnMS,kBpOgGnB,cAAL,GAAiB,GoOhGO,CpOmMT,CAAX,C;O;KoO1MzC,C;4FASA,yB;MAAA,wD;MpOsL  
A,0B;MAAA,+B;MoOtLA,4B;QAoOd,OpO0LX,eAAW,OoO1LQ,iBpOuFIB,cAAL,GAAiB,GoOvFM,CpO0LR,C  
AAX,C;O;KoOjMzC,C;gFAUA,yB;MAAA,4C;MpOqJA,+B;MoOrJA,sC;QAa2D,OpOkJjB,eoOIJ0B,WpOmD7B,c  
oOnD6B,EAAW,QAAX,CpOkJ1B,C;O;KoO/J1C,C;kFAeA,yB;MAAA,8C;MpOsIA,+B;MoOtIA,sC;QAa4D,OpO  
mIIB,eoOnI2B,YpOoC9B,coOpC8B,EAAAY,QAAX,CpOmI3B,C;O;KoOhJ1C,C;oFAeA,yB;MIogFI,6B;MkO3SJ,g  
D;MA2NA,4B;QAM+C,OA3NM,anO+RhB,CCeE,cAAU,cAAL,GAAiB,KAAtB,CDfF,MmO/RgB,C;O;KAqNrD,  
C;oGAQA,yB;MpHnLA,4B;MoHmLA,4B;QAMuD,OpHnLI,Y9GkNIB,c4N3CpC,GAAAY,K9GvK0C,C8GuKvD,G  
AA+C,EAA/C,I;O;KMMJ,C;sGAQA,yB;MNZA,kE;MMYA,4B;QAMwD,ONZoB,sB5NmCnC,c4NnCe,GAAW,K  
AAS,C;O;KMM5E,C;8FAQA,yB;MAAA,0D;MIouHA,4B;MAAA,iC;MkOvHA,4B;QAouD,OIO2HZ,gBAAY,Qk  
O3HQ,kBIOWBrB,cAAL,GAAiB,KkOxBS,CIO2HR,CAAZ,C;O;KkOII3C,C;4FASA,yB;MAAA,wD;MIO8GA,4B;  
MAAA,iC;MkO9GA,4B;QAosD,OIOkHX,gBAAY,QkOIHO,iBIOepB,cAAL,GAAiB,KkOfQ,CIOkHP,CAAZ,C;O  
;KkOzH3C,C;gFAUA,yB;MAAA,4C;MIOyFA,iC;MkOzFA,sC;QAa6D,OIOsFhB,gBkOtf0B,WIOX9B,ckOW8B,E  
AAW,QAAX,CIOsF1B,C;O;KkOnG7C,C;kFAeA,yB;MAAA,8C;MIO0EA,iC;MkO1EA,sC;QAa8D,OIOuEjB,gBk  
OvE2B,YIO1B/B,ckO0B+B,EAAAY,QAAX,CIOuE3B,C;O;KkOpF7C,C;ICtRA,qC;MAEL,SpOuIoD,coOvI3C,CpOu  
I2C,EoOvIvC,CpOuIuC,C;MoOtIpD,SpOsIoD,coOtI3C,CpOsI2C,EoOtvC,CpOsluC,C;MoOrIpD,OpOmDkE,YoO  
nDvD,EpOmDwE,KAAjB,EoOnDjD,EpOmD8E,KAA7B,CoOnDvD,KAAAX,GpOkFsD,SoOIFjC,EpOkF2C,KAAK  
,GoOIF3C,EpOkFuD,KAAZ,IAAf,CoOIFtD,GpOqEqD,SAAU,CAaT,SoOIFpB,EpOkF8B,KAAK,GoOIF9B,EpOk  
F0C,KAAZ,IAAf,CABs,MAAK,GoOrExB,CpOqEmC,KAAAX,IAAf,C;K;IoOIEzD,qC;MAEL,SnNwIsD,emNxI7C,  
CnNwI6C,EmNxIzC,CnNwIyC,C;MmNvItD,SnNuIsD,emNvI7C,CnNuI6C,EmNvIzC,CnNuIyC,C;MmNtItD,OnN  
qDmE,amNrDxD,EnNqD0E,KAAIB,EmNrDID,EnNqDgF,KAA9B,CmNrDxD,KAAAX,GnN+EWd,UmN/EnC,EnN  
+E8C,KAAK,UmN/E9C,EnN+E0D,KAAZ,CAAhB,CmN/ExD,GnNkEuD,UAAW,CAAv,UmN/EtB,EnN+EiC,KAA  
AK,UmN/EjC,EnN+E6C,KAAZ,CAAhB,CABU,MAAK,KmNIE3B,CnNkEsC,KAAAX,CAAhB,C;K;ImN/D3D,uD;  
MAmBI,WAAO,CAAP,C;QAD8E,OpOwBZ,YoOvBID,KpOuBmE,KAAjB,EoOvBzC,GpOuBsE,KAA7B,CoOvBI  
D,KAD8D,GACHd,GADgD,GpOuDxB,SoOtdf,GpOsDyB,KAAK,GoOtdxB,mBAAiB,GAAjB,EAAsB,KAAtB,E  
pO2WV,SoO3WuC,IpO2WvC,CoO3WU,CpOsDoC,KAAZ,IAAf,C;aoOrDtD,WAAO,CAAP,C;QAF8E,OpOwBZ,  
YoOtdID,KpOsBmE,KAAjB,EoOtdzC,GpOsBsE,KAA7B,CoOtdID,KAF8D,GAehD,GAfgD,GpO0CzB,SoOxCd  
,GpOwCwB,KAAK,GoOxCvB,mBAAiB,KAAjB,EAAwB,GAAxB,EpO0WV,SoO1WwC,CAAC,IAAD,IpO0WxC  
,CoO1WU,CpOwCkC,KAAAX,IAAf,C;QoOvC7C,MAAa,gCAAYB,eAAzB,C;K;IAGzB,uD;MAmBI,sBAAO,CAA  
P,C;QADkF,OnNQf,amNPnD,KnNOqE,KAAIB,EmNP1C,GnNOWE,KAA9B,CmNPnD,KADkE,GACpD,GADoD,  
GnNkC1B,UmNjCjB,GnNiC4B,KAAK,UmNjC3B,mBAAiB,GAAjB,EAAsB,KAAtB,EnNkWP,UmNIWoC,InNk  
WpC,CmNIWO,CnNiCuC,KAAZ,CAAhB,C;amNhCxD,sBAAO,CAAP,C;QAFkF,OnNQf,amNNnD,KnNMqE,KA  
AIB,EmNN1C,GnNMwE,KAA9B,CmNNnD,KAFkE,GAEPD,GAFOd,GnNqB3B,UmNnBhB,GnNmB2B,KAAK,  
KmNnB1B,mBAAiB,KAAjB,EAAwB,GAAxB,EnNiWP,UmNjWsC,IAAD,anNiWrC,CmNjWO,CnNmBqC,KAA  
X,CAAhB,C;QmNIB/C,MAAa,gCAAYB,eAAzB,C;K;InOIDC,sB;MAEtB,8B;MAFmG,gB;K;IAEnG,4B;MAAA,g



C;MAcI,iBAGqC,WAAO,CAAP,C;MAErC,iBAGqC,WAAO,MAAP,C;MAErC,kBAGmC,C;MAEnC,iBAGkC,E;K;;;IANtC,wC;MAAA,uC;QAAA,sB;;MAAA,gC;K;wGAsBA,iB;MAM0D,OAAa,0BA6OjC,SAAL,GAAiB,KA7OqB,EAAU,KF4O3C,KAAL,GAAiB,GE5OqB,C;K;oGAEvE,iB;MAOoE,OAAa,0BAoO3C,SAAL,GAAiB,KApO+B,EAAU,KAoOrD,KAAL,GAAiB,KApO+B,C;K;wGAEjF,yB;MA2PA,6B;MD5PA,8C;MCCA,wB;QAMyD,ODAS,YAAiB,CC8PhD,cAAU,SAAL,GAAiB,KAAtB,CD9PgD,MAAjB,ECAe,KDAc,KAA7B,C;O;KCNIE,C;wGAQA,yB;MA6PA,aAS6D,0B;MAT7D,+B;MgB9PA,gD;MhBCA,wB;QAM0D,OgBAS,aAAkB,ChBgQhD,eAAW,oBAAL,SAAK,CAAL,YAAN,CgBhQgD,MAAiB,EhBAGB,KgBAc,KAA9B,C;O;KhBNnE,C;8FAQA,yB;MA2OA,6B;MA3OA,wB;QAEsD,ODMD,cAAU,CC4O5B,cAAU,SAAL,GAAiB,KAAtB,CD5O4B,MAAK,GAAW,CD2O5C,cEjPsC,KFiP5B,KAAL,GAAiB,GAAtB,CC3O4C,MAAX,IAAf,C;O;KCRrD,C;8FAGA,yB;MAwOA,6B;MAxOA,wB;QAEuD,ODGF,cAAU,CC4O5B,cAAU,SAAL,GAAiB,KAAtB,CD5O4B,MAAK,GAAW,CC4O5C,cA/OuC,KA+O7B,KAAL,GAAiB,KAAtB,CD5O4C,MAAX,IAAf,C;O;KCLrD,C;8FAGA,yB;MAqOA,6B;MArOA,wB;QAEqD,ODAA,cAAU,CC4O5B,cAAU,SAAL,GAAiB,KAAtB,CD5O4B,MAAK,GCAI,KDAO,KAAZ,IAAf,C;O;KCFrD,C;8FAGA,yB;MA4OA,aAS6D,0B;MAT7D,+B;MA5OA,wB;QAEuD,OgBAA,eAAW,ChBmP7B,eAAW,oBAL,SAAK,CAAL,YAAN,CgBnP6B,MAAK,KhBAI,KgBAO,KAAZ,CAAhB,C;O;KhBFvD,C;gGAIA,yB;MA8NA,6B;MA9NA,wB;QAEuD,ODMD,cAAU,CC+N7B,cAAU,SAAL,GAAiB,KAAtB,CD/N6B,MAAK,GAAZ,CD8N9C,cEpOwC,KFoO9B,KAAL,GAAiB,GAAtB,CC9N8C,MAAZ,IAAf,C;O;KCRtD,C;gGAGA,yB;MA2NA,6B;MA3NA,wB;QAEwD,ODGF,cAAU,CC+N7B,cAAU,SAAL,GAAiB,KAAtB,CD/N6B,MAAK,GAAZ,CC+N9C,cAlOyC,KAKO/B,KAAL,GAAiB,KAAtB,CD/N8C,MAAZ,IAAf,C;O;KCLtD,C;gGAGA,yB;MAwNA,6B;MAxNA,wB;QAEsD,ODAA,cAAU,CC+N7B,cAAU,SAAL,GAAiB,KAAtB,CD/N6B,MAAK,GCAK,KDAO,KAAZ,IAAf,C;O;KCFtD,C;gGAGA,yB;MA+NA,aAS6D,0B;MAT7D,+B;MA/NA,wB;QAEwD,OgBAA,eAAW,ChBsO9B,eAAW,oBAAL,SAAK,CAAL,YAAN,CgBtO8B,MAAK,UhBAK,KgBAO,KAAZ,CAAhB,C;O;KhBFxD,C;gGAIA,yB;MAiNA,6B;MAjNA,wB;QAEuD,ODMD,cAAe,YAAL,CCkN7B,cAAU,SAAL,GAAiB,KAAtB,CDIN6B,MAAK,EAAY,CDiN9C,cEvNwC,KFuN9B,KAAL,GAAiB,GAAtB,CCjN8C,MAAZ,CAAf,C;O;KCRtD,C;gGAGA,yB;MA8MA,6B;MA9MA,wB;QAEwD,ODGF,cAAe,YAAL,CCkN7B,cAAU,SAAL,GAAiB,KAAtB,CDIN6B,MAAK,EAAY,CCKn9C,cArNyC,KaqN/B,KAAL,GAAiB,KAAtB,CDIN8C,MAAZ,CAAf,C;O;KCLtD,C;gGAGA,yB;MA2MA,6B;MA3MA,wB;QAEsD,ODAA,cAAe,YAAL,CCkN7B,cAAU,SAAL,GAAiB,KAAtB,CDIN6B,MAAK,ECAK,KDAO,KAAZ,CAAf,C;O;KCFtD,C;gGAGA,yB;MAkNA,aAS6D,0B;MAT7D,+B;MAINA,wB;QAEwD,OgBAA,eAAW,ChByN9B,eAAW,oBAAL,SAAK,CAAL,YAAN,CgBzN8B,MAAK,UhBAK,KgBAO,KAAZ,CAAhB,C;O;KhBFxD,C;4FAIA,yB;MAoMA,6B;MD9LA,4C;MCNA,wB;QAEqD,ODMD,WCqMjB,cAAU,SAAL,GAAiB,KAAtB,CDrMiB,EDoMjB,cE1MoC,KF0M1B,KAAL,GAAiB,GAAtB,CCpMiB,C;O;KCRpD,C;4FAGA,yB;MAiMA,6B;MD9LA,4C;MCHA,wB;QAEsD,ODGF,WCqMjB,cAAU,SAAL,GAAiB,KAAtB,CDrMiB,ECqMjB,cAxMqC,KAwM3B,KAAL,GAAiB,KAAtB,CDrMiB,C;O;KCLpD,C;4FAGA,yB;MA8LA,6B;MD9LA,4C;MCAA,wB;QAEoD,ODAA,WCqMjB,cAAU,SAAL,GAAiB,KAAtB,CDrMiB,ECAkB,KDAIB,C;O;KCFpD,C;4FAGA,yB;MAqMA,aAS6D,0B;MAT7D,+B;MgBrMA,8C;MhBAA,wB;QAEsD,OgBAA,YhB4MjB,eAAW,oBAAL,SAAK,CAAL,YAAN,CgB5MiB,EhBAmb,KgBAnB,C;O;KhBFtD,C;4FAIA,yB;MAuLA,6B;MDzKA,kD;MCdA,wB;QAMqD,ODcD,cC4KjB,cAAU,SAAL,GAAiB,KAAtB,CD5KiB,ED2KjB,cEzLoC,KFyL1B,KAAL,GAAiB,GAAtB,CC3KiB,C;O;KCPbD,C;4FAOA,yB;MAgLA,6B;MDzKA,kD;MCPA,wB;QAMsD,ODOF,cC4KjB,cAAU,SAAL,GAAiB,KAAtB,CD5KiB,EC4KjB,cAnLqC,KAmL3B,KAAL,GAAiB,KAAtB,CD5KiB,C;O;KCbD,C;4FAOA,yB;MAyKA,6B;MDzKA,kD;MCAA,wB;QAMoD,ODAA,cC4KjB,cAAU,SAAL,GAAiB,KAAtB,CD5KiB,ECAkB,KDAIB,C;O;KCNpD,C;4FAOA,yB;MA4KA,aAS6D,0B;MAT7D,+B;MgB5KA,oD;MhBAA,wB;QAMsD,OgBAA,ehB+KjB,eAAW,oBAAL,SAAK,CAAL,YAAN,CgB/KiB,EhBAmb,KgBAnB,C;O;KhBNtD,C;8GAQA,yB;MA0JA,6B;MD9LA,4C;MCoCA,wB;QAMiD,ODxCG,WCqMjB,cAAU,SAAL,GAAiB,KAAtB,CDrMiB,EDoMjB,cE5JqC,KF4J3B,KAAL,GAAiB,GAAtB,CCpMiB,C;O;KCKpD,C;sGAOA,yB;MAmJA,6B;MD9LA,4C;MC2CA,wB;QAMkD,OD/CE,WCqMjB,cAAU,SAAL,GAAiB,KAAtB,CDrMiB,ECqMjB,cAtJsC,KAsJ5B,KAAL,GAAiB,KAAtB,CDrMiB,C;O;KCyCpD,C;sGAOA,yB;MA4IA,6B;MD9LA,4C;MCKDA,wB;QAMgD,ODtDI,WCqMjB,cAAU,SAAL,GAAiB,KAAtB,CDrMiB,ECsDmB,KDtDnB,C;O;KCGDpD,C;sGAOA,yB;MA+IA,aAS6D,0B;MAT7D,+B;MgBrMA,8C;MhBsDA,wB;QAMkD,OgB1DI,YhB4MjB,eAAW,oBAAL,SAAK,CAAL,YAAN,CgB5MiB,EhB0DoB,KgB1DpB,C;O;KhBoDtD,C;4FAQA,yB;MA6HA,6B;MDzKA,kD;MDuOJ,0B;MAAA,+B;ME3LI,wB;QAQ6C,OF8LR,eAAW,

OC5OI,cC4KjB,cAAU,SAAL,GAAiB,KAAtB,CD5KiB,ED2KjB,cE7H4B,KF6HIB,KAAL,GAAiB,GAAtB,CC3KiB,CAkLf,KD0DW,CAAX,C;O;KEtMrC,C;4FASA,yB;MAoHA,6B;MDzKA,kD;MCwOJ,4B;MAAA,iC;MAnLI,wB;QAQ+C,OAsLR,gBAAY,QD7OC,cC4KjB,cAAU,SAAL,GAAiB,KAAtB,CD5KiB,EC4KjB,cArH8B,KAqHpB,KAAL,GAAiB,KAAtB,CD5KiB,CA4Lb,KCiDY,CAAZ,C;O;KA9LvC,C;4FASA,yB;MA2GA,6B;MDzKA,kD;MC8DA,wB;QAQ2C,ODhES,cC4KjB,cAAU,SAAL,GAAiB,KAAtB,CD5KiB,ECgES,KDhET,C;O;KCwDpD,C;4FASA,yB;MA4GA,aAS6D,0B;MAT7D,+B;MgB5KA,oD;MhBgEA,wB;QAQ6C,OgBIES,ehB+KjB,eAAW,oBAAL,SAAK,CAAL,YAAN,CgB/KiB,EhBkEU,KgBIEV,C;O;KhB0DtD,C;4EAUA,yB;MAAA,4B;MAAA,iC;MAAA,mB;QAM2C,uBAAY,QAAL,SAAK,KAAZ,C;O;KAN3C,C;4EAQA,yB;MAAA,4B;MAAA,iC;MAAA,mB;QAM2C,uBAAY,QAAL,SAAK,KAAZ,C;O;KAN3C,C;oGAQA,yB;MAAA,8C;MAwEA,6B;MAxEA,wB;QAE+D,0BA+E5B,cAAU,SAAL,GAAiB,KAAtB,CA/E4B,EA+E5B,cA/EqD,KA+E3C,KAAL,GAAiB,KAAtB,CA/E4B,C;O;KAF/D,C;4FAIA,yB;MAAA,iC;MyLnNJ,4B;MzLmNI,wB;QAEqD,uByLiNiC,QzLkN1B,IAAK,KyLINX,GzLkNoB,KAAM,KyLINM,CzLkNjC,C;O;KAFrD,C;0FAGA,yB;MAAA,iC;MyLjNJ,4B;MzLiNI,wB;QAEoD,uByLhNgC,QzLgNzB,IAAK,KyLhNX,GzLgNmB,KAAM,KyLhNM,CzLgNhC,C;O;KAFpD,C;4FAGA,yB;MAAA,iC;MyL/MJ,4B;MzL+MI,wB;QAEqD,uByL9MiC,QzL8M1B,IAAK,KyL9MX,GzL8MoB,KAAM,KyL9MM,CzL8MjC,C;O;KAFrD,C;4EAGA,yB;MAAA,iC;MyL7MJ,4B;MzL6MI,mB;QAEkC,uByL5MsB,QAAP,CzL4MR,SyL5Me,CzL4MtB,C;O;KAFIC,C;kFAIA,yB;MAAA,0B;MAAA,mB;QAUMC,OAAK,OAAL,SAAK,C;O;KAVxC,C;oFAWA,Y;MASqC,gB;K;gFACrC,Y;MASiC,OAAK,SAAL,GAAiB,K;K;kFACID,yB;MAAA,aASqD,0B;MATrD,mB;QASmC,OAAK,oBAAL,SAAK,CAAL,Y;O;KATnC,C;oFAWA,yB;MF+DJ,0B;MAAA,+B;ME/DI,mB;QASqC,OFiEE,eAAW,OEjEb,SFiEa,CAAX,C;O;KE1EvC,C;sFAUA,Y;MAEuC,W;K;kFACvC,yB;MAAA,6B;MAAA,mB;QASmC,qBAAU,SAAL,GAAiB,KAAtB,C;O;KATnC,C;oFAUA,yB;MAAA,aAS6D,0B;MAT7D,+B;MAAA,mB;QASqC,sBAAW,oBAAL,SAAK,CAAL,YAAN,C;O;KATrC,C;oFAWA,Y;MAMqC,OApDC,SAAL,GAAiB,K;K;sFAqDID,Y;MAMuC,OA3DD,SAAL,GAAiB,K;K;gCA6DID,Y;MAAYC,OAAQ,CA7DX,SAAL,GAAiB,KA6DD,Y;K;:::;gCA3UrD,Y;MAAA,c;MAGuG,qD;MAHvG,a;K;8BAAA,iB;MAAA,2IAGuG,oCAHvG,G;K;0EA+UA,yB;MAAA,iC;MAAA,4B;QAW4C,uBAAY,SAAZ,C;O;KAX5C,C;4EAYA,yB;MAAA,iC;MAAA,4B;QAU6C,uBAAO,SAAP,C;O;KAV7C,C;4EAWA,yB;MAAA,4B;MAAA,iC;MAAA,4B;QAW2C,uBAAY,QAAL,SAAK,CAAZ,C;O;KAX3C,C;4EAYA,yB;MAAA,4B;MAAA,iC;MAAA,4B;QAW4C,uBAAY,QAAL,SAAK,SAAZ,C;O;KAX5C,C;IkC/WA,8B;MACqB,sB;K;wCAKjB,iB;MAM8C,OICsVL,WkCtVK,aAAQ,KAAR,CICsVL,C;K;wCkCpVzC,wB;MAOI,aAAQ,KAAR,IAAiB,KIC4OgB,K;K;mFkCxOP,Y;MAAQ,OAAA,YAAQ,O;K;qCAE9C,Y;MAC+E,gCAAS,YAAT,C;K;IAEzD,qC;MAAC,oB;MACnB,eAAoB,C;K;6CACpB,Y;MAAYB,sBAAQ,YAAM,O;K;0CACvC,Y;MAAoD,Q;MAA9B,IAAI,eAAQ,YAAM,OAAIB,C;QAAA,OICiUe,WkCjUS,aAAM,mBAAN,EAAM,2BAAN,OICiUt,C;;QkCjUwC,MAAM,2BAAuB,YAAM,WAA7B,C;K;;2CAGvF,mB;MAIS,Q;MAAL,IAAI,eAAC,0EAAD,SAAJ,C;QAAkC,OAAO,K;MAEzC,OAAe,WAAR,YAAQ,EAAS,OICuNS,KkCvNIB,C;K;gDAGnB,oB;MACY,Q;MAA2B,gBAA3B,gE;MAA2B,c;;QjB6nDvB,U;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,6B;QAAhB,OAAgB,gBAAhB,C;UAAgB,2B;UiB7nD6B,2BjB6nDR,OiB7nDQ,S;UAAA,W;YAAwB,oBAAR,YAAQ,EjB6nDhC,OjB16CA,KkCnNgC,C;;UjB6nD/C,IAAI,OAAJ,C;YAAyB,aAAO,K;YAAP,e;;QAC/C,aAAO,I;;MiB9nDH,iB;K;oCAGJ,Y;MAAkC,OAAA,IAAK,QAAQ,OAAb,KAAqB,C;K;;IA9CvD,uC;MAAA,qD;MACgC,wBAAK,eAAW,IAAX,CAAL,C;MADhC,Y;K;:::;qCAPJ,Y;MAAA,OAKqB,sDALrB,M;K;qCAA,Y;MAAA,c;MAKqB,wD;MALrB,a;K;mCAA,iB;MAAA,2IAKqB,0CALrB,G;K;kFAwDA,yB;MAAA,2C;MAWwC,0C;QAAA,wB;UAAW,OAAA,aAAK,KAAL,CICkMV,K;S;O;MkC7MzC,6B;QAWI,OAAO,qBAAY,gCAAW,IAAX,GAAiB,wBAAjB,CAAZ,C;O;KAXX,C;oFAcA,oB;MAGwE,e;K;IkM3ExE,sC;MAQ2D,OAAa,WAAb,StOwQjB,KAAL,GAAiB,GsOxQkB,EAAS,KAAT,C;K;IAExE,sC;MAQ4D,OAAa,WAAb,SpO+PIB,KAAL,GAAiB,KoO/PmB,EAAS,KAAT,C;K;IAGzE,sC;MAQ0D,OAAc,WrOiR5B,oBqOjRc,SrOiRnB,KAAK,CAAL,iBqOjRiC,EAAS,KAAT,C;K;IAExE,sC;MAOGd,uBAAc,SpNyQvB,KoNzQS,EAA6B,WAAW,KAAX,CAA7B,C;K;IAGhD,8B;MAMqC,Q;MAAA,0DAAmB,kBAAkB,SAAlB,C;K;IAExD,qC;MAO+C,Q;MAAA,0CAAc,KAAd,oBAAwB,kBAAkB,SAAlB,C;K;IAGvE,+B;MAMuC,Q;MAAA,2DAAoB,kBAAkB,SAAlB,C;K;IAE3D,sC;MAOiD,Q;MAAA,2CAAE,KAAf,oBAAyB,kBAAkB,SAAlB,C;K;IAE1E,6B;MAMmC,Q;MAAA,yDAAkB,kBAAkB,SAAlB,C;K;IAErD,oC;MAO6C,Q;MAAA,yCAAa,KAAb,oBAAuB,kBAAkB,SAAlB,C;K;IAEpE,8B;MAMqC,Q;MAAA,0DAAmB,kBAAkB,SAAlB,C;K;IAExD,qC;MAO+C,Q;MAAA,0CAAc,KAAd,oBAAwB,kBAAkB,SAAlB,C;K;IAMvE,kC;MAM4C,kCA

AsB,EAAtB,C;K;IAE5C,2C;MASmB,Q;MAAA,sBAAL,SAAK,EAAa,KAAb,C;MAAL,iB;QAA4B,OAAO,I;MAA7C,UAAU,I;MACV,IrO/EkE,YqO+E9D,GrO/E+E,KAAjB,EAA6B,CD6P5D,SsO9KzB,6BAAM,UtO8K6B,KAA L,GAAiB,GAAtB,CC7P4D,MAA7B,CqO+E9D,IAAJ,C;QAA2B,OAAO,I;MAClC,OtO8OqC,UAAW,OsO9OzC,G rOoL8B,KD0DW,CAAX,C;K;IsO3OzC,mC;MAM8C,mCAAuB,EAAvB,C;K;IAE9C,4C;MASmB,Q;MAAA,sBA AL,SAAK,EAAa,KAAb,C;MAAL,iB;QAA4B,OAAO,I;MAA7C,UAAU,I;MACV,IrOrGkE,YqOqG9D,GrOrG+E, KAAjB,EAA6B,CC8P5D,SoOzJzB,8BAAO,UpOyJ4B,KAAL,GAAiB,KAAtB,CD9P4D,MAA7B,CqOqG9D,IAAJ ,C;QAA4B,OAAO,I;MACnC,OpOyNuC,WAAy,QoOzN5C,GrOwKgC,KCiDY,CAAZ,C;K;IoOtN3C,iC;MAM0C, iCAAqB,EAArB,C;K;IAE1C,0C;MASI,WAAW,KAAX,C;MAEA,aAAa,SAAK,O;MACIB,IAAI,WAAU,CAAd,C; QAAiB,OAAO,I;MAExB,YAAkB,4BAAK,U;MACvB,S;MAEA,gBAAgB,qBAAK,CAAL,C;MACHB,IAAI,YAA Y,EAAhB,C;QACI,IAAI,WAAU,CAAV,IAAe,cAAa,EAAhC,C;UAAqC,OAAO,I;QAC5C,QAAQ,C;QAER,QAA Q,C;MAGZ,uBAAuB,mB;MAEvB,qBAAqB,gB;MACrB,arOuMmC,SqOvMtB,KrOuMsB,C;MqOtMnC,aAAa,W; MACb,aAAU,KAAV,MAAsB,MAAtB,M;QACI,YAAy,QAAQ,qBAAK,CAAL,CAAR,EAAiB,KAAjB,C;QAEZ,I AAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,IrOnJ8D,YqOmJ1D,MrOnJ2E,KAAjB,EqOmJjD,crOnJ8E,KAA7B, CqOmJ1D,IAAJ,C;UACI,IAAI,+CAAkB,gBAAIB,QAAJ,C;YACI,iBrO5FwC,WqO4FvB,KrO5FuB,EqO4Ff,MrO5 Fe,C;YqO8FxC,IrOvJsD,YqOuJlD,MrOvJmE,KAAjB,EqOuJzC,crOvJsE,KAA7B,CqOuJlD,IAAJ,C;cACI,OAAO,I ;;YAGX,OAAO,I;QAlf,SrOnHkD,SAAE,YqOmHjE,MrOnH4D,KAAK,EqOmHvD,MrOnHmE,KAAZ,CAAF,C; QqOqHlD,mBAAmB,M;QACnB,SrOhJiD,SqOgJjD,MrOhJ2D,KAAK,GAAW,CAkU5C,SqOILrB,KrOkLqB,CAI U4C,MAAX,IAAf,C;QqOiJjD,IrOnK8D,YqOmK1D,MrOnK2E,KAAjB,EqOmKjD,YrOnK8E,KAA7B,CqOmK1D ,IAAJ,C;UAA2B,OAAO,I;MAGtC,OAAO,M;K;IAGX,kC;MAM4C,kCAAsB,EAAtB,C;K;IAE5C,2C;MASI,WA AW,KAAX,C;MAEA,aAAa,SAAK,O;MACIB,IAAI,WAAU,CAAd,C;QAAiB,OAAO,I;MAExB,YAAmB,6BAA M,U;MACzB,S;MAEA,gBAAgB,qBAAK,CAAL,C;MACHB,IAAI,YAAy,EAAhB,C;QACI,IAAI,WAAU,CAAV,I AAE,cAAa,EAAhC,C;UAAqC,OAAO,I;QAC5C,QAAQ,C;QAER,QAAQ,C;MAIZ,uBAAuB,gD;MAEvB,qBAAq B,gB;MACrB,apN0lqC,UAAW,oBoN1InC,KpN0ImC,CAAX,C;MoNzIrC,aAAa,2B;MACb,aAAU,KAAV,MAAs B,MAAtB,M;QACI,YAAy,QAAQ,qBAAK,CAAL,CAAR,EAAiB,KAAjB,C;QAEZ,IAAI,QAAQ,CAAZ,C;UAAe ,OAAO,I;QACtB,IpN5M+D,aoN4M3D,MpN5M6E,KAAIB,EoN4MID,cpN5MgF,KAA9B,CoN4M3D,IAAJ,C;UA CI,IAAI,+CAAkB,gBAAIB,QAAJ,C;YACI,iBpN1J0C,YoN0JzB,KpN1JyB,EoN0JjB,MpN1JiB,C;YoN4J1C,IpNh NuD,aoNgNnD,MpNhNqE,KAAIB,EoNgN1C,cpNhNwE,KAA9B,CoNgNnD,IAAJ,C;cACI,OAAO,I;YAGX,OA AO,I;QAlf,SpNjLoD,UoNiLpD,MpNjL+D,KAAK,UoNiL1D,MpNjLsE,KAAZ,CAAhB,C;QoNmLpD,mBAAmB, M;QACnB,SpN9MmD,UoN8MnD,MpN9M8D,KAAK,KAAW,CjBsQ7C,UAAW,oBAAL,CAyDR,SqOjHrB,KrOi HqB,CAzDQ,MAAK,CAAL,iBAAN,CiBtQ6C,MAAX,CAAhB,C;QoN+MnD,IpN5N+D,aoN4N3D,MpN5N6E,KA AIB,EoN4NID,YpN5NgF,KAA9B,CoN4N3D,IAAJ,C;UAA2B,OAAO,I;MAGtC,OAAO,M;K;I7N9RX,6B;MACK D,OAAuB,0BAAtB,KAAO,WAAe,EAAU,KAAO,WAAjB,C;K;IACzE,8B;MACqD,OAAC,gCAAuB,iBAAU,gC AAV,C;K;IAE7E,4B;MACoD,ORiZZ,SAvGI,oBQ1SS,ER0Sd,KAAK,CAAL,iBQ1Sc,KR0ST,oBQ1SuB,ER0S5B, KAAK,CAAL,iBQ1Sc,CRiZH,QAAV,C;K;IQhZxC,+B;MACuD,OR+Yf,SAvGI,oBQxSY,ERwSjB,KAAK,CAAL, iBQxSiB,QRwSZ,oBQxS0B,ERwS/B,KAAK,CAAL,iBQxSiB,CR+YN,QAAV,C;K;IQ1YxC,6B;MAEI,eAAe,ESk SoB,K;MTjSnC,cAAc,ESiSqB,K;MThSnC,IAAI,qBAAU,CAAd,C;QACI,OS6C+D,aT7CpD,ES6CsE,KAAIB,ET7 C/C,ES6C6E,KAA9B,CT7CpD,IAAJ,GAAa,aAAb,GAA2B,a;MAItC,IAAI,uBAAY,CAAhB,C;QACI,OAAO,UA AM,aAAW,OAAX,CAAN,C;MAIX,eAAiB,4BAAC,CAAd,CAAD,KAAoB,OAAPB,CAAD,WAAkC,CAAIC,C; MACf,UAAU,kBAAW,kBAAW,OAAX,CAAX,C;MACV,OAAO,UAAM,iCSkCsD,aAAkB,CTICzD,UAAM,GA AN,CSkCyD,MAAIB,EAA8B,CTICvD,UAAM,OAAN,CSkCuD,MAA9B,CTICvC,KAAJ,GAAkC,CAAIC,GAAy C,CAAPD,EAAN,C;K;IAIX,gC;MAKe,Q;MAHX,eAAe,ES8QoB,K;MT7QnC,cAAc,ES6QqB,K;MT5QnC,IAAI,q BAAU,CAAd,C;QACW,ISyBwD,aTzBpD,ESyBsE,KAAIB,ETzB/C,ESyB6E,KAA9B,CTzBpD,IAAJ,C;UACH,S; UAEA,OSgDgD,UThDhD,ESgD2D,KAAK,UThD3D,ESgDuE,KAAZ,CAAhB,C;QTnDpD,W;MAQJ,IAAI,uBA AY,CAAhB,C;QACI,OAAO,UAAM,gBAAW,OAAX,CAAN,C;MAIX,eAAiB,4BAAC,CAAd,CAAD,KAAoB,O AAPB,CAAD,WAAkC,CAAIC,C;MACf,UAAU,kBAAW,kBAAW,OAAX,CAAX,C;MACV,OAAO,UAAM,aSUs D,aAAkB,CTV9D,UAAM,GAAN,CSU8D,MAAIB,EAA8B,CTV5D,UAAM,OAAN,CSU4D,MAA9B,CTV5C,KA AJ,GAAkC,OAAC,KAAN,CAAN,C;K;IAGX,yB;MAEI,IAAE,QAAF,CAAE,CAAF,C;QADyC,OAC5B,W;QAC b,SRwSuC,aQxSiC,4BAAK,URwS0C,KAAb,CQxSvC,C;UAFyC,OAEP,4BAAK,U;UACvC,SRuSuC,aQvSiC,4B



# 1.37 project-lombok 1.18.20

## 1.37.1 Available under license :

Copyright (C) 2009-2021 The Project Lombok Authors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

Licenses for included components:

org.ow2.asm:asm

org.ow2.asm:asm-analysis

org.ow2.asm:asm-commons

org.ow2.asm:asm-tree

org.ow2.asm:asm-util

ASM: a very small and fast Java bytecode manipulation framework

Copyright (c) 2000-2011 INRIA, France Telecom

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-----  
rzwitserloot/com.zwitsersloot.cmdreader

Copyright 2010 Reinier Zwitsersloot.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-----  
rzwitserloot/lombok.patcher

Copyright (C) 2009-2021 The Project Lombok Authors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-----

## 1.38 thanhpk/randstr v1.0.4

### 1.38.1 Available under license :

The MIT License

Copyright (c) 2010-2018 Google, Inc. <http://angularjs.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.39 aws-sdk-for-java 2.17.122

### 1.39.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of



the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

#### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang  
Copyright 2001-2020 The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (<https://www.apache.org/>).

## 1.40 smartystreets-assertions v1.2.1

### 1.40.1 Available under license :

```
This is the official list of people who can contribute
(and typically have contributed) code to the go-diff
repository.
#
The AUTHORS file lists the copyright holders; this file
lists people. For example, ACME Inc. employees would be listed here
but not in AUTHORS, because ACME Inc. would hold the copyright.
#
```

# When adding J Random Contributor's name to this file,  
# either J's name or J's organization's name should be  
# added to the AUTHORS file.  
#  
# Names should be added to this file like so:  
# Name <email address>  
#  
# Please keep the list sorted.

Danny Yoo <dannyoo@google.com>  
James Kolb <jkolb@google.com>  
Jonathan Amsterdam <jba@google.com>  
Markus Zimmermann <markus.zimmermann@nethead.at> <markus.zimmermann@symflower.com>  
<zimmski@gmail.com>  
Matt Kovars <akaskik@gmail.com>  
rjan Persson <orjan@spotify.com>  
Osman Masood <oamasood@gmail.com>  
Robert Carlsen <rwcarlsen@gmail.com>  
Rory Flynn <roryflynn@users.noreply.github.com>  
Sergi Mansilla <sergi.mansilla@gmail.com>  
Shatrugna Sadhu <ssadhu@apcera.com>  
Shawn Smith <shawnpsmith@gmail.com>  
Stas Maksimov <maksimov@gmail.com>  
Tor Arvid Lund <torarvid@gmail.com>  
Zac Bergquist <zbergquist99@gmail.com>  
Copyright (c) 2012-2016 The go-diff Authors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2016 SmartyStreets, LLC

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

NOTE: Various optional and subordinate components carry their own licensing requirements and restrictions. Use of those components is subject to the terms and conditions outlined the respective license of each component.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions



for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
// Copyright (c) 2015 The Chromium Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
```

// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT  
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,  
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT  
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,  
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY  
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT  
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE  
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.41 coocood-freecache v1.2.0

### 1.41.1 Available under license :

The MIT License

Copyright (c) 2015 Ewan Chou.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.42 guava-listenablefuture-only 9999.0-empty-to-avoid-conflict-with-guava

### 1.42.1 Available under license :

Found license 'GNU Lesser General Public License' in '// This library is free software; you can redistribute it and/or  
// modify it under the terms of the GNU Lesser General Public // License as published by the Free Software  
Foundation; either // version 2.1 of the License, or (at your option) any later version. // This library is distributed in  
the hope that it will be useful, // but WITHOUT ANY WARRANTY; without even the implied warranty of //  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU // Lesser General Public  
License for more details. // You should have received a copy of the GNU Lesser General Public \* This grammar is  
in the PUBLIC DOMAIN'

## GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts  
as the successor of the GNU Library Public License, version 2, hence  
the version number 2.1.]

### Preamble

The licenses for most software are designed to take away your  
freedom to share and change it. By contrast, the GNU General Public  
Licenses are intended to guarantee your freedom to share and change  
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some  
specially designated software packages--typically libraries--of the  
Free Software Foundation and other authors who decide to use it. You  
can use it too, but we suggest you first think carefully about whether  
this license or the ordinary General Public License is the better  
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,  
not price. Our General Public Licenses are designed to make sure that  
you have the freedom to distribute copies of free software (and charge  
for this service if you wish); that you receive source code or can get  
it if you want it; that you can change the software and use pieces of  
it in new free programs; and that you are informed that you can do  
these things.

To protect your rights, we need to make restrictions that forbid  
distributors to deny you these rights or to ask you to surrender these  
rights. These restrictions translate to certain responsibilities for  
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis  
or for a fee, you must give the recipients all the rights that we gave  
you. You must make sure that they, too, receive or can get the source  
code. If you link other code with the library, you must provide  
complete object files to the recipients, so that they can relink them  
with the library after making changes to the library and recompiling  
it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the

library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

## GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from

such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in

themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the



Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under

Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on

the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to

refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing

and reuse of software generally.

## NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU  
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public  
License along with this library; if not, write to the Free Software  
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your  
school, if any, to sign a "copyright disclaimer" for the library, if  
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the  
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!

```
////////////////////////////////////
// checkstyle: Checks Java source code for adherence to a set of rules.
// Copyright (C) 2001-2020 the original author or authors.
//
// This library is free software; you can redistribute it and/or
// modify it under the terms of the GNU Lesser General Public
// License as published by the Free Software Foundation; either
// version 2.1 of the License, or (at your option) any later version.
//
// This library is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
// Lesser General Public License for more details.
//
// You should have received a copy of the GNU Lesser General Public
// License along with this library; if not, write to the Free Software
// Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
////////////////////////////////////
```

## 1.43 re2j 1.5

### 1.43.1 Available under license :

Copyright (c) \${year} The Go Authors. All rights reserved.

Use of this source code is governed by a BSD-style

license that can be found in the LICENSE file.

This is a work derived from Russ Cox's RE2 in Go, whose license <http://golang.org/LICENSE> is as follows:

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.44 pkgerrors v0.9.1

### 1.44.1 Available under license :

Copyright (c) 2015, Dave Cheney <dave@cheney.net>  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice,

this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.45 scala v2.12.10

### 1.45.1 Available under license :

Scala includes the JLine library:

Copyright (c) 2002-2006, Marc Prud'hommeaux <mwp1@cornell.edu>  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of JLine nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED



AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
{% if site.thisScalaVersion != site.latestScalaVersion %}
<div class="version-notice">This is the specification of a previous version of Scala. See the <a href="{ {
site.baseurl} }"/>{ { site.latestScalaVersion } }"/">Scala { { site.latestScalaVersion } } spec.</div>
{% endif %}
```

```
class C {
 val x: ((Int, Int) => Int) = (((a, b) => a)
 val y: ((Int, Int, Int) => Int) = (((a, !!)) => a)
 val z: ((Int, Int, Int) => Int) = (((a, NotAPatternVariableName, c)) => a)
}
```

Scala

Copyright (c) 2002-2019 EPFL

Copyright (c) 2011-2019 Lightbend, Inc.

Scala includes software developed at  
LAMP/EPFL (<https://lamp.epfl.ch/>) and  
Lightbend, Inc. (<https://www.lightbend.com/>).

Licensed under the Apache License, Version 2.0 (the "License").  
Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

This software includes projects with other licenses -- see `doc/LICENSE.md`.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,  
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by  
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all  
other entities that control, are controlled by, or are under common  
control with that entity. For the purposes of this definition,  
"control" means (i) the power, direct or indirect, to cause the  
direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

Scala includes the ASM library.

Copyright (c) 2000-2011 INRIA, France Telecom  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright

notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

not-a-legal-formal-parameter-tuple.scala:2: error: not a legal formal parameter.

Note: Tuples cannot be directly destructured in method or function parameters.

```
 Either create a single parameter accepting the Tuple2,
 or consider a pattern matching anonymous function: ` { case (a, b) => ... }
val x: ((Int, Int) => Int) = (((a, b)) => a)
 ^
```

not-a-legal-formal-parameter-tuple.scala:3: error: not a legal formal parameter.

Note: Tuples cannot be directly destructured in method or function parameters.

```
 Either create a single parameter accepting the Tuple2,
 or consider a pattern matching anonymous function: ` { case (param1, param2) => ... }
val y: ((Int, Int, Int) => Int) = (((a, !)) => a)
 ^
```

not-a-legal-formal-parameter-tuple.scala:4: error: not a legal formal parameter.

Note: Tuples cannot be directly destructured in method or function parameters.

```
 Either create a single parameter accepting the Tuple3,
 or consider a pattern matching anonymous function: ` { case (param1, ..., param3) => ... }
val z: ((Int, Int, Int) => Int) = (((a, NotAPatternVariableName, c)) => a)
 ^
```

three errors found

Scala is licensed under the [Apache License Version 2.0](<https://www.apache.org/licenses/LICENSE-2.0>).

## Scala License

Copyright (c) 2002-2019 EPFL

Copyright (c) 2011-2019 Lightbend, Inc.

All rights reserved.

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

#### # Other Licenses

This software includes projects with the following licenses,  
which are also included in the `licenses/`` directory:

### [Apache License](<http://www.apache.org/licenses/LICENSE-2.0.html>)

This license is used by the following third-party libraries:

- \* jansi

### [BSD License](<http://www.opensource.org/licenses/bsd-license.php>)

This license is used by the following third-party libraries:

- \* jline

### [BSD 3-Clause License](<http://opensource.org/licenses/BSD-3-Clause>)

This license is used by the following third-party libraries:

- \* asm

### [MIT License](<http://www.opensource.org/licenses/MIT>)

This license is used by the following third-party libraries:

- \* jquery

- \* tools tooltip

package scala.build

```
import sbt._, Keys._, plugins._
```

```
object License extends AutoPlugin {
 val licenseMapping = settingKey[Seq[(File, String)]]("LICENSE/NOTICE file mappings")
```

```
 override val requires = JvmPlugin
```

```
 override val trigger = AllRequirements
```

```
override def projectSettings: Seq[Def.Setting[_]] =
 List(packageSrc, packageBin, packageDoc)
 .map(pkg => mappings in (Compile, pkg) += licenseMapping.value)

override def buildSettings: Seq[Def.Setting[_]] = Seq(
 licenseMapping := List("LICENSE", "NOTICE").map(fn => (baseDirectory.value / fn) -> fn)
)
}
```

Scala includes the Tools Tooltip library:

Copyright (c) 2009 Tero Piirainen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

(c) 2012-2014 GitHub

When using the GitHub logos, be sure to follow the GitHub logo guidelines (<https://github.com/logos>)

Font License: SIL OFL 1.1 (<http://scripts.sil.org/OFL>)

Applies to all font files

Code License: MIT (<http://choosealicense.com/licenses/mit/>)

Applies to all other files

Copyright (c) 2006, Ivan Sagalaev

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of highlight.js nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE



DISCLAIMED. IN NO EVENT SHALL THE REGENTS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Scala includes the JLine library, which includes the Jansi library.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed

with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Scala includes the jQuery library:

Copyright (c) 2010 John Resig

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

(The MIT License)

Copyright (c) 2013 Greg Allen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE

## 1.46 jetbrains-kotlin-kotlin-stdlib-jdk7 1.7.20

### 1.46.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2010-2017 JetBrains s.r.o.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
*/opt/cola/permits/1526005753_1673461516.6385698/0/kotlin-stdlib-jdk7-1-7-20-sources-
jar/kotlin/AutoCloseable.kt
```

## 1.47 everit-json-schema 1.12.2

### 1.47.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<!--
```

Copyright (C) 2011 Everit Kft. (<http://www.everit.org>)

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and

limitations under the License.

-->

Found in path(s):

\* /opt/cola/permits/1411867003\_1662683754.0293086/0/everit-json-schema-1-12-2-sources-2-jar/META-INF/maven/com.github.erosb/everit-json-schema/pom.xml

No license file was found, but licenses were detected in source scan.

/\*

Copyright (c) 2002 JSON.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The Software shall be used for Good, not Evil.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

\*/

Found in path(s):

\* /opt/cola/permits/1411867003\_1662683754.0293086/0/everit-json-schema-1-12-2-sources-2-jar/org/everit/json/schema/JSONPointer.java

\* /opt/cola/permits/1411867003\_1662683754.0293086/0/everit-json-schema-1-12-2-sources-2-jar/org/everit/json/schema/JSONPointerException.java

No license file was found, but licenses were detected in source scan.

/\*

Copyright (c) 2006 JSON.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The Software shall be used for Good, not Evil.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

\*/

Found in path(s):

\* /opt/cola/permits/1411867003\_1662683754.0293086/0/everit-json-schema-1-12-2-sources-2-jar/org/everit/json/schema/internal/JSONWriter.java

## 1.48 netty/resolver/dns/classes/macOS

### 4.1.85.Final

#### 1.48.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<!--
~ Copyright 2021 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ https://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.
-->
```

Found in path(s):

\* /opt/cola/permits/1498798544\_1670284052.8406405/0/netty-resolver-dns-classes-macos-4-1-85-final-sources-2-jar/META-INF/maven/io.netty/netty-resolver-dns-classes-macos/pom.xml



No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2019 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1498798544_1670284052.8406405/0/netty-resolver-dns-classes-macos-4-1-85-final-sources-2-
jar/io/netty/resolver/dns/macos/package-info.java
* /opt/cola/permits/1498798544_1670284052.8406405/0/netty-resolver-dns-classes-macos-4-1-85-final-sources-2-
jar/io/netty/resolver/dns/macos/DnsResolver.java
* /opt/cola/permits/1498798544_1670284052.8406405/0/netty-resolver-dns-classes-macos-4-1-85-final-sources-2-
jar/io/netty/resolver/dns/macos/MacOSDnsServerAddressStreamProvider.java
```

## 1.49 junit 4.13.2

### 1.49.1 Available under license :

JUnit

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

#### 1. DEFINITIONS

"Contribution" means:

- a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
- b) in the case of each subsequent Contributor:
  - i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

## 2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights

needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

### 3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

### 4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

## 5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

## 6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST

PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## 7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this



\* 6. Redistributions of any form whatsoever must retain the following  
 \* acknowledgment:  
 \* "This product includes software developed by the OpenSSL Project  
 \* for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"  
 \*  
 \* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS" AND ANY  
 \* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
 \* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR  
 \* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR  
 \* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,  
 \* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT  
 \* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;  
 \* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
 \* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,  
 \* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)  
 \* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED  
 \* OF THE POSSIBILITY OF SUCH DAMAGE.  
 \* =====  
 \*  
 \* This product includes cryptographic software written by Eric Young  
 \* (eay@cryptsoft.com). This product includes software written by Tim  
 \* Hudson (tjh@cryptsoft.com).  
 \*  
 \*/

Original SSLeay License

-----

/\* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)  
 \* All rights reserved.  
 \*  
 \* This package is an SSL implementation written  
 \* by Eric Young (eay@cryptsoft.com).  
 \* The implementation was written so as to conform with Netscapes SSL.  
 \*  
 \* This library is free for commercial and non-commercial use as long as  
 \* the following conditions are aheared to. The following conditions  
 \* apply to all code found in this distribution, be it the RC4, RSA,  
 \* lhash, DES, etc., code; not just the SSL code. The SSL documentation  
 \* included with this distribution is covered by the same copyright terms  
 \* except that the holder is Tim Hudson (tjh@cryptsoft.com).  
 \*  
 \* Copyright remains Eric Young's, and as such any Copyright notices in  
 \* the code are not to be removed.  
 \* If this package is used in a product, Eric Young should be given attribution  
 \* as the author of the parts of the library used.  
 \* This can be in the form of a textual message at program startup or  
 \* in documentation (online or textual) provided with the package.

- \*
  - \* Redistribution and use in source and binary forms, with or without
  - \* modification, are permitted provided that the following conditions
  - \* are met:
    - \* 1. Redistributions of source code must retain the copyright
    - \* notice, this list of conditions and the following disclaimer.
    - \* 2. Redistributions in binary form must reproduce the above copyright
    - \* notice, this list of conditions and the following disclaimer in the
    - \* documentation and/or other materials provided with the distribution.
    - \* 3. All advertising materials mentioning features or use of this software
    - \* must display the following acknowledgement:
      - \* "This product includes cryptographic software written by
      - \* Eric Young (eay@cryptsoft.com)"
      - \* The word 'cryptographic' can be left out if the routines from the library
      - \* being used are not cryptographic related :-).
    - \* 4. If you include any Windows specific code (or a derivative thereof) from
    - \* the apps directory (application code) you must include an acknowledgement:
      - \* "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
- \*
  - \* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS" AND
  - \* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
  - \* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
  - \* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
  - \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
  - \* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
  - \* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
  - \* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
  - \* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
  - \* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
  - \* SUCH DAMAGE.
- \*
  - \* The licence and distribution terms for any publically available version or
  - \* derivative of this code cannot be changed. i.e. this code cannot simply be
  - \* copied and put under another distribution licence
  - \* [including the GNU Public Licence.]

\*/  
 BoringSSL is a fork of OpenSSL. As such, large parts of it fall under OpenSSL licensing. Files that are completely new have a Google copyright and an ISC license. This license is reproduced at the bottom of this file.

Contributors to BoringSSL are required to follow the CLA rules for Chromium:  
<https://cla.developers.google.com/cla>

Files in third\_party/ have their own licenses, as described therein. The MIT license, for third\_party/ fiat, which, unlike other third\_party directories, is compiled into non-test libraries, is included below.

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact



openssl-core@openssl.org.

The following are Google-internal bug numbers where explicit permission from some authors is recorded for use of their work. (This is purely for our own record keeping.)

27287199

27287880

27287883

OpenSSL License

-----

/\* =====

\* Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.

\*

\* Redistribution and use in source and binary forms, with or without  
\* modification, are permitted provided that the following conditions  
\* are met:

\*

\* 1. Redistributions of source code must retain the above copyright  
\* notice, this list of conditions and the following disclaimer.

\*

\* 2. Redistributions in binary form must reproduce the above copyright  
\* notice, this list of conditions and the following disclaimer in  
\* the documentation and/or other materials provided with the  
\* distribution.

\*

\* 3. All advertising materials mentioning features or use of this  
\* software must display the following acknowledgment:  
\* "This product includes software developed by the OpenSSL Project  
\* for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"

\*

\* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to  
\* endorse or promote products derived from this software without  
\* prior written permission. For written permission, please contact  
\* [openssl-core@openssl.org](mailto:openssl-core@openssl.org).

\*

\* 5. Products derived from this software may not be called "OpenSSL"  
\* nor may "OpenSSL" appear in their names without prior written  
\* permission of the OpenSSL Project.

\*

\* 6. Redistributions of any form whatsoever must retain the following  
\* acknowledgment:  
\* "This product includes software developed by the OpenSSL Project  
\* for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"

\*

\* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY  
\* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
\* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR  
\* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR  
\* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

\* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT  
\* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;  
\* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
\* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,  
\* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)  
\* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED  
\* OF THE POSSIBILITY OF SUCH DAMAGE.

\* =====

\*

\* This product includes cryptographic software written by Eric Young  
\* (eay@cryptsoft.com). This product includes software written by Tim  
\* Hudson (tjh@cryptsoft.com).

\*

\*/

Original SSLeay License

-----

/\* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)

\* All rights reserved.

\*

\* This package is an SSL implementation written

\* by Eric Young (eay@cryptsoft.com).

\* The implementation was written so as to conform with Netscapes SSL.

\*

\* This library is free for commercial and non-commercial use as long as

\* the following conditions are aheared to. The following conditions

\* apply to all code found in this distribution, be it the RC4, RSA,

\* lhash, DES, etc., code; not just the SSL code. The SSL documentation

\* included with this distribution is covered by the same copyright terms

\* except that the holder is Tim Hudson (tjh@cryptsoft.com).

\*

\* Copyright remains Eric Young's, and as such any Copyright notices in

\* the code are not to be removed.

\* If this package is used in a product, Eric Young should be given attribution

\* as the author of the parts of the library used.

\* This can be in the form of a textual message at program startup or

\* in documentation (online or textual) provided with the package.

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\* 1. Redistributions of source code must retain the copyright

\* notice, this list of conditions and the following disclaimer.

\* 2. Redistributions in binary form must reproduce the above copyright

\* notice, this list of conditions and the following disclaimer in the

\* documentation and/or other materials provided with the distribution.

\* 3. All advertising materials mentioning features or use of this software

\* must display the following acknowledgement:

\* "This product includes cryptographic software written by

\* Eric Young (eay@cryptsoft.com)"  
\* The word 'cryptographic' can be left out if the rouines from the library  
\* being used are not cryptographic related :-).  
\* 4. If you include any Windows specific code (or a derivative thereof) from  
\* the apps directory (application code) you must include an acknowledgement:  
\* "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"  
\*

\* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS" AND  
\* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
\* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
\* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE  
\* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
\* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
\* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
\* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT  
\* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY  
\* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
\* SUCH DAMAGE.

\*  
\* The licence and distribution terms for any publically available version or  
\* derivative of this code cannot be changed. i.e. this code cannot simply be  
\* copied and put under another distribution licence  
\* [including the GNU Public Licence.]

\*/

ISC license used for completely new code in BoringSSL:

/\* Copyright (c) 2015, Google Inc.

\*

\* Permission to use, copy, modify, and/or distribute this software for any  
\* purpose with or without fee is hereby granted, provided that the above  
\* copyright notice and this permission notice appear in all copies.

\*

\* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES  
\* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF  
\* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY  
\* SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES  
\* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION  
\* OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN  
\* CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. \*/

The code in third\_party/ fiat carries the MIT license:

Copyright (c) 2015-2016 the fiat-crypto authors (see

<https://github.com/mit-plv/fiat-crypto/blob/master/AUTHORS>).

Permission is hereby granted, free of charge, to any person obtaining a copy  
of this software and associated documentation files (the "Software"), to deal  
in the Software without restriction, including without limitation the rights  
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
copies of the Software, and to permit persons to whom the Software is  
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all

copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Licenses for support code

-----

Parts of the TLS test suite are under the Go license. This code is not included in BoringSSL (i.e. libcrypto and libssl) when compiled, however, so distributing code linked against BoringSSL does not trigger this license:

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

BoringSSL uses the Chromium test infrastructure to run a continuous build, trybots etc. The scripts which manage this, and the script for generating build metadata, are under the Chromium license. Distributing code linked against BoringSSL does not trigger this license.

Copyright 2015 The Chromium Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above

copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but

not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.  
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.



## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The Netty Project

=====

Please visit the Netty web site for more information:

\* <http://netty.io/>

Copyright 2016 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

-----

This product contains a forked and modified version of Tomcat Native

- \* LICENSE:
  - \* license/LICENSE.tomcat-native.txt (Apache License 2.0)
- \* HOMEPAGE:
  - \* <http://tomcat.apache.org/native-doc/>
  - \* <https://svn.apache.org/repos/asf/tomcat/native/>

This product contains the Maven wrapper scripts from 'Maven Wrapper', that provides an easy way to ensure a user has everything necessary to run the Maven build.

- \* LICENSE:
  - \* license/LICENSE.mvn-wrapper.txt (Apache License 2.0)
- \* HOMEPAGE:
  - \* <https://github.com/takari/maven-wrapper>

This product contains small piece of code to support AIX, taken from netbsd.

- \* LICENSE:
  - \* license/LICENSE.aix-netbsd.txt (OpenSSL License)
- \* HOMEPAGE:
  - \* <https://ftp.netbsd.org/pub/NetBSD/NetBSD-current/src/crypto/external/bsd/openssl/dist>

This product contains code from boringssl.

- \* LICENSE (Combination ISC and OpenSSL license)
  - \* license/LICENSE.boringssl.txt (Combination ISC and OpenSSL license)
- \* HOMEPAGE:
  - \* <https://boringssl.googlesource.com/boringssl/>

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
  
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
  
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or,

within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all

other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.51 apache-commons-digester 2.1

## 1.51.1 Available under license :

Apache Commons Digester

Copyright 2001-2010 The Apache Software Foundation

This product includes software developed by  
The Apache Software Foundation (<http://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,

the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:



- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
  
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
  
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

## 1.52 bean-validation-api 2.0.1

### 1.52.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,  
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by  
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all  
other entities that control, are controlled by, or are under common  
control with that entity. For the purposes of this definition,  
"control" means (i) the power, direct or indirect, to cause the  
direction or management of such entity, whether by contract or  
otherwise, or (ii) ownership of fifty percent (50%) or more of the  
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity  
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,  
including but not limited to software source code, documentation  
source, and configuration files.

"Object" form shall mean any form resulting from mechanical  
transformation or translation of a Source form, including but

not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.  
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2013 Cognifide Limited

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.53 goconvey v1.7.2

## 1.53.1 Available under license :

Copyright (c) 2016 SmartyStreets, LLC

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

NOTE: Various optional and subordinate components carry their own licensing requirements and restrictions. Use of those components is subject to the terms and conditions outlined the respective license of each component.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the



editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the

same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## 1.54 mockito-junit-jupiter 3.11.2

### 1.54.1 Available under license :

The MIT License

Copyright (c) 2007 Mockito contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.55 duct-tape 1.0.8

## 1.55.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Richard North

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.56 leaktest v1.3.0

### 1.56.1 Available under license :

Copyright (c) 2012 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.57 json-java 20201115

### 1.57.1 Available under license :

Copyright (c) 2002 JSON.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The Software shall be used for Good, not Evil.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.58 kotlin-reflect 1.7.20

### 1.58.1 Available under license :

Apache-2.0

## 1.59 prometheus-common v0.32.1

### 1.59.1 Available under license :

Common libraries shared by Prometheus Go components.

Copyright 2015 The Prometheus Authors

This product includes software developed at

SoundCloud Ltd. (<http://soundcloud.com/>).

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and



- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## 1.60 javax-annotation-api 1.3.2

### 1.60.1 Available under license :

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

#### 1. Definitions.

1.1. Contributor. means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version. means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable. means the Covered Software in any form other than Source Code.

1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License. means this document.

1.8. Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or .Your.) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, .You. includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, .control. means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

## 2. License Grants.

### 2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

### 2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor

either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

### 3. Distribution Obligations.

#### 3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

#### 3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

#### 3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

#### 3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

#### 3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or

alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

### 3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

## 4. Versions of the License.

### 4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

### 4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

### 4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

## 5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN .AS IS. BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

## 6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms

herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as .Participant.) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

## 7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY.S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

## 8. U.S. GOVERNMENT END USERS.

The Covered Software is a .commercial item., as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of .commercial computer software. (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and .commercial computer software documentation. as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

## 9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original

Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

#### 10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

#### NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.



To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or

binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

#### Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

#### "CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular source file's header the words

"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that

accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

## 1.61 google-gson 2.10

### 1.61.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0  
Created-By: 17.0.4.1 (Oracle Corporation)  
Build-Jdk-Spec: 17  
Bnd-LastModified: 1666660169762  
Bundle-ContactAddress: <https://github.com/google/gson>  
Bundle-Description: Gson JSON library  
Bundle-DocURL: <https://github.com/google/gson/gson>  
Bundle-License: "Apache-2.0";link="https://www.apache.org/licenses/LICENSE-2.0.txt"  
Bundle-ManifestVersion: 2  
Bundle-Name: Gson  
Bundle-RequiredExecutionEnvironment: JavaSE-1.7, JavaSE-1.8  
Bundle-SCM: url="https://github.com/google/gson/gson",connection="scm:git:https://github.com/google/gson.git/gson",developer-connection="scm:git:git@github.com:google/gson.git/gson",tag="gson-parent-2.10"  
Bundle-SymbolicName: com.google.gson  
Bundle-Vendor: Google Gson Project  
Bundle-Version: 2.10.0  
Export-Package: com.google.gson;uses:="com.google.gson.reflect,com.google.gson.stream";version="2.10.0",com.google.gson.annotations;version="2.10.0",com.google.gson.reflect;version="2.10.0",com.google.gson.stream;version="2.10.0"  
Import-Package: sun.misc;resolution:=optional,com.google.gson.annotations  
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.7))"  
Tool: Bnd-6.3.1.202206071316  
Multi-Release: true

Found in path(s):

# 1.62 kubernetes-apimachinery 20191123-snapshot-4c4803ed

## 1.62.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.



4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed

with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.63 aws-java-sdk-::-services-::-amazon-dynamodb 2.17.101

## 1.63.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by  
Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](https://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:  
Apache Commons Lang  
Copyright 2001-2020 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).

# 1.64 aws-sdk-for-java 2.17.101

## 1.64.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.



"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by  
Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-

2010 AOL Inc.

- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:  
Apache Commons Lang  
Copyright 2001-2020 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).

## 1.65 kotlin-scripting-common 1.7.20

### 1.65.1 Available under license :

Apache-2.0

## 1.66 aws-java-sdk-::-services-::-aws-glue

### 2.17.122

### 1.66.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the

outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and

do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.



9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

## THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:  
Apache Commons Lang  
Copyright 2001-2020 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).

# 1.67 junit-jupiter-junit-jupiter-api 5.8.2

## 1.67.1 Available under license :

Eclipse Public License - v 2.0

=====

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

### ### 1. Definitions

Contribution means:

\* \*\*a)\*\* in the case of the initial Contributor, the initial content Distributed under this Agreement, and

\* \*\*b)\*\* in the case of each subsequent Contributor:

\* \*\*i)\*\* changes to the Program, and

\* \*\*ii)\*\* additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of **\*\*a)\*\*** distributing or **\*\*b)\*\*** making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

## ### 2. Grant of Rights

**\*\*a)\*\*** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

**\*\*b)\*\*** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

**\*\*c)\*\*** Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

**\*\*d)\*\*** Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

**\*\*e)\*\*** Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

### ### 3. Requirements

**\*\*3.1)\*\*** If a Contributor Distributes the Program in any form, then:

**\*\*a)\*\*** the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

**\*\*b)\*\*** the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

**\*\*i)\*\*** effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

**\*\*ii)\*\*** effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

**\*\*iii)\*\*** does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

**\*\*iv)\*\*** requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

**\*\*3.2)\*\*** When the Program is Distributed as Source Code:

**\*\*a)\*\*** it must be made available under this Agreement, or if the Program **\*\*i)\*\*** is combined with other material in a separate file or files made available under a Secondary License, and **\*\*ii)\*\*** the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

**\*\*b)\*\*** a copy of this Agreement must be included with each copy of the Program.

**\*\*3.3)\*\*** Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

### ### 4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: **\*\*a)\*\*** promptly

notify the Commercial Contributor in writing of such claim, and \*\*b)\*\* allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

#### ### 5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

#### ### 6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### ### 7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this

Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

#### #### Exhibit A - Form of Secondary Licenses Notice

> This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here }.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Open Source Licenses

=====

This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the subcomponent's license, as noted in the LICENSE-<subcomponent>.md files.

## 1.68 jackson-dataformats-binary 2.14.0

### 1.68.1 Available under license :

No license file was found, but licenses were detected in source scan.

<p class="legalCopy"><small>Copyright &#169; 2022 <a href="http://fasterxml.com/">FasterXML</a>. All rights reserved.</small></p>

Found in path(s):

- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/CBORParser.Feature.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/class-use/CBORConstants.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/overview-tree.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/class-use/CBORGenerator.Feature.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/constant-values.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/CBORFactoryBuilder.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/CBORConstants.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/class-use/CBORGenerator.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/databind/class-use/CBORMapper.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/databind/class-use/CBORMapper.Builder.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/class-use/CBORSimpleValue.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/index-all.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/overview-summary.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/class-use/CBORReadContext.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/class-use/CBORFactoryBuilder.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/databind/CBORMapper.Builder.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/databind/package-summary.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/CBORSimpleValue.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/package-use.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/class-use/CBORFactory.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/CBORGenerator.Feature.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/CBORParserBootstrapper.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/class-use/CBORWriteContext.html
- \* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-

jar/com/fasterxml/jackson/dataformat/cbor/databind/CBORMapper.html  
\* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/package-summary.html  
\* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/deprecated-list.html  
\* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/help-doc.html  
\* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/serialized-form.html  
\* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/databind/package-tree.html  
\* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/databind/package-use.html  
\* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/CBORParser.html  
\* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/PackageVersion.html  
\* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/CBORWriteContext.html  
\* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/class-use/CBORParserBootstrapper.html  
\* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/package-tree.html  
\* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/class-use/CBORParser.html  
\* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/CBORGenerator.html  
\* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/class-use/PackageVersion.html  
\* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/class-use/CBORParser.Feature.html  
\* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/CBORReadContext.html  
\* /opt/cola/permits/1473601746\_1668802746.968431/0/jackson-dataformat-cbor-2-14-0-javadoc-jar/com/fasterxml/jackson/dataformat/cbor/CBORFactory.html

# 1.69 apache-httpcomponents-core 4.4.13

## 1.69.1 Available under license :

Apache HttpComponents Core  
Copyright 2005-2020 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<http://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>



## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner

or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. **Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

## 1.70 handy-uri-templates handy-uri-templates-2.1.8

### 1.70.1 Available under license :

No license file was found, but licenses were detected in source scan.

<url><http://www.apache.org/licenses/LICENSE-2.0.txt></url>

Found in path(s):

\* /opt/cola/permits/1168630323\_1662743882.4324734/0/handy-uri-templates-2-1-8-jar/META-INF/maven/com.damnhandy/handy-uri-templates/pom.xml

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Automatic-Module-Name: com.damnhandy.uri.template

Bnd-LastModified: 1558834247596

Build-Jdk: 11.0.3  
Built-By: ryan  
Bundle-Description: Handy URI Templates is a RFC6570 compliant URI template processor. The library allows clients to utilize templated URIs and inject replacement variables to expand the template into a URI. The library sports a fluent API, ability to plugin custom object renderers, and supports all levels of URI templates.  
Bundle-DocURL: <https://github.com/damnhandy/Handy-URI-Templates>  
Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>  
Bundle-ManifestVersion: 2  
Bundle-Name: handy-uri-templates  
Bundle-SymbolicName: com.damnhandy.handy-uri-templates  
Bundle-Vendor: Ryan J. McDonough  
Bundle-Version: 2.1.8  
Created-By: Apache Maven Bundle Plugin  
Export-Package: com.damnhandy.uri.template;version="2.1.8",com.damnhandy.uri.template.jackson.datatype;uses:="com.damnhandy.uri.template,com.fasterxml.jackson.core,com.fasterxml.jackson.databind,com.fasterxml.jackson.databind.module";version="2.1.8"  
Import-Package: com.fasterxml.jackson.core;version="[2.9,3)";resolution:=optional,com.fasterxml.jackson.databind;version="[2.9,3)";resolution:=optional,com.fasterxml.jackson.databind.module;version="[2.9,3)";resolution:=optional,org.joda.time;version="[2.10,3)",org.joda.time.format;version="[2.10,3)"  
key: value  
mode: development  
Require-Capability: osgi.ee;filter:="(&(osgi.ee=JavaSE)(version=1.7))"  
Tool: Bnd-4.2.0.201903051501  
url: <https://github.com/damnhandy/Handy-URI-Templates>

Found in path(s):

\* /opt/cola/permits/1168630323\_1662743882.4324734/0/handy-uri-templates-2-1-8-jar/META-INF/MANIFEST.MF

# 1.71 jetbrains-kotlin-kotlin-scripting-compiler-embeddable 1.3.50

## 1.71.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of

the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.



8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

#### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

## 1.72 jmespath-go-jmespath v0.4.0

### 1.72.1 Available under license :

Copyright 2015 James Saryerwinnie

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## 1.73 jackson-annotations 2.14.0

### 1.73.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
  
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.74 snakeyaml-engine 2.5

## 1.74.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<name>Apache License, Version 2.0</name>
<url>http://www.apache.org/licenses/LICENSE-2.0.txt</url>
```

Found in path(s):

```
* /opt/cola/permits/1526005986_1673041495.0443194/0/snakeyaml-engine-2-5-jar/META-INF/maven/org.snakeyaml/snakeyaml-engine/pom.xml
```

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0  
Automatic-Module-Name: org.snakeyaml.engine.v2  
Bnd-LastModified: 1664826555266  
Build-Jdk-Spec: 1.8  
Bundle-Description: Core YAML 1.2 parser and emitter for Java  
Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>  
Bundle-ManifestVersion: 2  
Bundle-Name: SnakeYAML Engine  
Bundle-SymbolicName: org.snakeyaml.engine  
Bundle-Version: 2.5.0  
Created-By: Apache Maven Bundle Plugin  
Export-Package: org.snakeyaml.engine.v2.api;version="2.5",org.snakeyaml.engine.v2.api.lowlevel;version="2.5",org.snakeyaml.engine.v2.comments;version="2.5",org.snakeyaml.engine.v2.common;version="2.5",org.snakeyaml.engine.v2.composer;version="2.5",org.snakeyaml.engine.v2.constructor;version="2.5",org.snakeyaml.engine.v2.emitter;version="2.5",org.snakeyaml.engine.v2.env;version="2.5",org.snakeyaml.engine.v2.events;version="2.5",org.snakeyaml.engine.v2.exceptions;version="2.5",org.snakeyaml.engine.v2.nodes;version="2.5",org.snakeyaml.engine.v2.parser;version="2.5",org.snakeyaml.engine.v2.representer;version="2.5",org.snakeyaml.engine.v2.resolver;version="2.5",org.snakeyaml.engine.v2.scanner;version="2.5",org.snakeyaml.engine.v2.serializer;version="2.5",org.snakeyaml.engine.v2.tokens;version="2.5"  
Import-Package: org.snakeyaml.engine.v2.api;version="[2.5,3)",org.snakeyaml.engine.v2.comments;version="[2.5,3)",org.snakeyaml.engine.v2.composer;version="[2.5,3)",org.snakeyaml.engine.v2.constructor;version="[2.5,3)",org.snakeyaml.engine.v2.emitter;version="[2.5,3)",org.snakeyaml.engine.v2.env;version="[2.5,3)",org.snakeyaml.engine.v2.events;version="[2.5,3)",org.snakeyaml.engine.v2.exceptions;version="[2.5,3)",org.snakeyaml.engine.v2.nodes;version="[2.5,3)",org.snakeyaml.engine.v2.parser;version="[2.5,3)",org.snakeyaml.engine.v2.representer;version="[2.5,3)",org.snakeyaml.engine.v2.resolver;version="[2.5,3)",org.snakeyaml.engine.v2.scanner;version="[2.5,3)",org.snakeyaml.engine.v2.serializer;version="[2.5,3)",org.snakeyaml.engine.v2.tokens;version="[2.5,3)"  
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"  
Tool: Bnd-5.1.1.202006162103

Found in path(s):

\* /opt/cola/permits/1526005986\_1673041495.0443194/0/snakeyaml-engine-2-5-jar/META-INF/MANIFEST.MF

## 1.75 assertj-fluent-assertions 3.22.0

## 1.75.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance
 * with
 * the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
 * an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the
 * specific language governing permissions and limitations under the License.
 *
 * Copyright 2012-2021 the original author or authors.
 */
/**
 * Base class for all implementations of assertions for { @link File}s.
 *
 * @param <SELF> the "self" type of this assertion class. Please read "<a href="http://bit.ly/1IZIRcY"
 * target="_blank">Emulating 'self types' using Java Generics to simplify fluent API
 * implementation"
 * for more details.
 *
 * @author David DIDIER
 * @author Yvonne Wang
 * @author Alex Ruiz
 * @author Olivier Michallat
 * @author Olivier Demeijer
 * @author Mikhail Mazursky
 * @author Jean-Christophe Gay
 * @author Valeriy Vyrva
 * @author Nikolaos Georgiou
 */
```

Found in path(s):

```
* /opt/cola/permits/1341130270_1654817020.9922328/0/assertj-core-3-22-0-sources-
jar/org/assertj/core/api/AbstractFileAssert.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance
 * with
 * the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *

```



\* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on  
\* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the  
\* specific language governing permissions and limitations under the License.

\*  
\* Copyright 2012-2021 the original author or authors.

\*/  
/\*\*

\* Base class for all implementations of assertions for { @link Float}s.

\*  
\* @param <SELF> the "self" type of this assertion class. Please read &quot;<a href="http://bit.ly/1IZIRcY"  
\* target="\_blank">Emulating 'self types' using Java Generics to simplify fluent API  
implementation</a>&quot;  
\* for more details.

\*  
\* @author Drummond Dawson  
\* @author Yvonne Wang  
\* @author Alex Ruiz  
\* @author Ansgar Konermann  
\* @author Mikhail Mazursky  
\* @author Nicolas François  
\* @author Jin Kwon  
\*/

Found in path(s):

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-  
jar/org/assertj/core/api/AbstractFloatAssert.java

No license file was found, but licenses were detected in source scan.

/\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance  
with

\* the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on  
\* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the

\* specific language governing permissions and limitations under the License.

\*

\* Copyright 2012-2021 the original author or authors.

\*/

/\*\*

\* Base class for all implementations of assertions for { @code CharSequence}s.

\*

\* @param <SELF> the "self" type of this assertion class. Please read &quot;<a href="http://bit.ly/1IZIRcY"  
\* target="\_blank">Emulating 'self types' using Java Generics to simplify fluent API

implementation</a>&quot;

- \* for more details.
- \* @param <ACTUAL> the type of the "actual" value.
- \*
- \* @author Yvonne Wang
- \* @author David DIDIER
- \* @author Alex Ruiz
- \* @author Joel Costigliola
- \* @author Mikhail Mazursky
- \* @author Nicolas Francois
- \* @author Daniel Weber
- \*/

Found in path(s):

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractCharSequenceAssert.java

No license file was found, but licenses were detected in source scan.

/\*

- \* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with
- \* the License. You may obtain a copy of the License at
- \*
- \* <http://www.apache.org/licenses/LICENSE-2.0>
- \*
- \* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
- \* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- See the License for the
- \* specific language governing permissions and limitations under the License.
- \*
- \* Copyright 2012-2021 the original author or authors.
- \*/

/\*\*

- \* Base class for all implementations of assertions for { @link Short}s.
- \*
- \* @param <SELF> the "self" type of this assertion class. Please read &quot;<a href="http://bit.ly/1IZIRcY" target="\_blank">Emulating 'self types' using Java Generics to simplify fluent API
- implementation</a>&quot;
- \* for more details.
- \*
- \* @author Drummond Dawson
- \* @author Yvonne Wang
- \* @author David DIDIER
- \* @author Ansgar Konermann
- \* @author Alex Ruiz
- \* @author Mikhail Mazursky
- \* @author Nicolas François
- \* @author Cal027

\*/

Found in path(s):

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractShortAssert.java

No license file was found, but licenses were detected in source scan.

/\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

\* the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on

\* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

\* specific language governing permissions and limitations under the License.

\*

\* Copyright 2012-2021 the original author or authors.

\*/

/\*\*

\* Base class for all implementations of assertions for { @link Double}s.

\*

\* @param <SELF> the "self" type of this assertion class. Please read ["http://bit.ly/1IZIRcY"](http://bit.ly/1IZIRcY)

\* `target="_blank">Emulating 'self types' using Java Generics to simplify fluent API`

implementation</a>"

\* `for more details.`

\*

\* @author Drummond Dawson

\* @author Yvonne Wang

\* @author David DIDIER

\* @author Alex Ruiz

\* @author Ansgar Konermann

\* @author Joel Costigliola

\* @author Mikhail Mazursky

\* @author Nicolas François

\* @author Jack Gough

\*/

Found in path(s):

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractDoubleAssert.java

No license file was found, but licenses were detected in source scan.

/\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

```
* the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the
* specific language governing permissions and limitations under the License.
*
* Copyright 2012-2021 the original author or authors.
*/
/**
* Base class for all implementations of assertions for { @link List}s.
* @param <SELF> the "self" type of this assertion class. Please read "<a href="http://bit.ly/1IZIRcY"
* target="_blank">Emulating 'self types' using Java Generics to simplify fluent API
implementation"
* for more details.
* @param <ACTUAL> the type of the "actual" value.
* @param <ELEMENT> the type of elements of the "actual" value.
* @param <ELEMENT_ASSERT> used for navigational assertions to return the right assert type.
*
* @author Yvonne Wang
* @author Alex Ruiz
* @author Joel Costigliola
* @author Mikhail Mazursky
* @author Jacek Jackowiak
*/
```

Found in path(s):

```
* /opt/cola/permits/1341130270_1654817020.9922328/0/assertj-core-3-22-0-sources-
jar/org/assertj/core/api/AbstractListAssert.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance
with
* the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the
* specific language governing permissions and limitations under the License.
*
* Copyright 2012-2021 the original author or authors.
*/
/**
```

```
* Base class for all implementations of assertions for { @link Byte}s.
*
* @param <SELF> the "self" type of this assertion class. Please read "<a href="http://bit.ly/1IZIRcY"
* target="_blank">Emulating 'self types' using Java Generics to simplify fluent API
implementation";
* for more details.
*
*
* @author Drummond Dawson
* @author Yvonne Wang
* @author David DIDIER
* @author Ansgar Konermann
* @author Alex Ruiz
* @author Mikhail Mazursky
* @author Nicolas François
* @author Cal027
*/
```

Found in path(s):

```
* /opt/cola/permits/1341130270_1654817020.9922328/0/assertj-core-3-22-0-sources-
jar/org/assertj/core/api/AbstractByteAssert.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance
with
* the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the
* specific language governing permissions and limitations under the License.
*
* Copyright 2012-2021 the original author or authors.
*/
```

```
/**
* Base class for all implementations of assertions for { @link InputStream}s.
* @param <SELF> the "self" type of this assertion class. Please read "<a href="http://bit.ly/1IZIRcY"
* target="_blank">Emulating 'self types' using Java Generics to simplify fluent API
implementation";
* for more details.
* @param <ACTUAL> the type of the "actual" value.
*
* @author Matthieu Baechler
* @author Mikhail Mazursky
* @author Stefan Birkner
*/
```

Found in path(s):

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractInputStreamAssert.java

No license file was found, but licenses were detected in source scan.

/\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

\* the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

\* specific language governing permissions and limitations under the License.

\*

\* Copyright 2012-2021 the original author or authors.

\*/

Found in path(s):

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/FutureAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveAllNullFields.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/recursive/comparison/RecursiveComparisonConfiguration.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/AssertionErrorCreator.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Java6Assertions.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotContainKey.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/OffsetDateTimeByInstantComparator.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ClassAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotMatch.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractCompletableFutureAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ErrorMessageFactory.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/AbstractComparisonStrategy.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveSameClass.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractSpliteratorAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Files.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/recursive/comparison/FieldComparators.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotContainSequence.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveBinaryContent.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AssertionErrorCollector.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeUpperCase.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ElementsShouldBeExactly.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotHaveSameHashCode.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/description/Description.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Int2DArrays.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeOdd.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeMarked.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveNoParent.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/OptionalDoubleAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/introspection/FieldUtils.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/uri/ShouldHaveNoHost.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractIntArrayAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeAtSameInstant.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/CommonValidations.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainExactlyInAnyOrder.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AtomicIntegerArrayAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/OptionalLongAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/condition/Join.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveParent.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractIteratorAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/extractor/Extractors.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/TextFileWriter.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainPattern.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/description/TextDescription.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AtomicLongArrayAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldOnlyHaveFields.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/recursive/comparison/DualValueDeque.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/NioFilesWrapper.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/UnsatisfiedRequirement.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldExist.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Boolean2DArrays.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/IterableElementComparisonStrategy.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/JUnitBDDSoftAssertions.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Urls.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AssertionInfo.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Fail.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/WholeNumbers.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeRelativePath.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/uri/ShouldHaveQuery.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/BDDAssumptions.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotBeInstance.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/introspection/Introspection.java



\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/uri/ShouldHavePath.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Maps.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/junit/jupiter/SoftAssertionsExtension.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractArrayAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotBeEqualComparingFieldByFieldRecursively.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ObjectAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractInstantAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/IterableSizeAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Byte2DArrayAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeToday.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ElementsShouldSatisfy.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/CharacterAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Iterables.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ElementsShouldHaveExactly.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeBlank.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/EnumerableAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotBeInfinite.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/diff/DiffAlgorithm.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainCharSequence.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractUrlAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/ComparatorBasedComparisonStrategy.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/MapSizeAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHavePropertyOrFieldWithValue.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHavePeriod.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveName.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractPredicateAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEqualWithTimePrecision.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/BigDecimalAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Float2DArrayAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveDimensions.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveOnlyElementsOfType.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/introspection/MethodSupport.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/introspection/PropertySupport.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/BDDAssertions.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractDurationAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveNoSuperclass.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/BDDSoftAssertions.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/recursive/comparison/ComparisonKeyDifference.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/InputStreamAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/future/ShouldBeCancelled.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeInSameHour.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeInfinite.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeUnmodifiable.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/UriAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainOnlyDigits.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractOptionalAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Assertions.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/OptionalDoubleShouldHaveValueCloseToPercentage.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeAnArray.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotHaveDuplicates.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractEnumerableAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/IterableAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainSequenceOfCharSequence.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/FloatAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEqualIgnoringNewLineDifferences.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ArraySortedAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Short2DArrays.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEqualIgnoringNewLines.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotBeFinite.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ZippedElementsShouldSatisfy.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Objects.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/BooleanArrayAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ElementsShouldHaveAtLeast.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeReadable.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainExactly.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/matcher/AssertionMatcher.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/AbstractShouldHaveTextContent.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeNormalized.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeRegularFile.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/configuration/Configuration.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeCanonicalPath.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/recursive/comparison/RecursiveComparisonDifferenceCalculator.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/ChronoLocalDateTimeComparator.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/ClassNameComparator.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeInterface.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/CharArrayAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeBeforeYear.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveSameHashCode.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/AssertionErrorMessagesAggregator.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/Closeables.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/condition/DoesNotHave.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/DefaultAssertionErrorCollector.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveRootCause.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotBeInstanceOfAny.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Float2DArrays.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotHave.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/NotThrownAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Booleans.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AssertionsForInterfaceTypes.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/condition/AllOf.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/DurationAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Boolean2DArrayAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEqualNormalizingUnicode.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/LongAdderAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveSameHourAs.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/array2d/Array2dElementShouldBeDeepEqual.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeBefore.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractBooleanArrayAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/BasicErrorMessageFactory.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/IntArrayAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeAbstract.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/WithThrowable.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveMessageFindingMatchRegex.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeInSameHourWindow.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveNext.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/SoftProxies.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEqualToIgnoringFields.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveCauseReference.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveRootCauseInstance.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/LocalDateAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/future/ShouldBeCompletedExceptionally.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/WithAssertions.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeInTheFuture.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/future/ShouldHaveFailedWithin.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/filter/InFilter.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotContainCharSequence.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Doubles.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeSame.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEven.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotHaveToString.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainKeys.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Array2DAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/condition/MappedCondition.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/presentation/UnicodeRepresentation.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContain.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveSuppressedException.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/presentation/StandardRepresentation.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/BigIntegerAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/Arrays.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeInSameMonth.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/LocalTimeAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ClassBasedNavigableIterableAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/description/JoinDescription.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotEndWith.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/DateUtil.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeCloseTo.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/data/Offset.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/GenericComparableAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/SoftAssertionsStatement.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveRootCauseExactlyInstance.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Long2DArrays.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/DigestDiff.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveExtension.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Uris.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/iterable/ThrowingExtractor.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainAnyOf.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Double2DArrayAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveAnnotations.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeSubstring.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ListAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/BinaryDiffResult.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/diff/myers/Snake.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveCauseExactlyInstance.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/NoElementsShouldSatisfy.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotContainAnyWhitespaces.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ShortArrayAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEqualWithinPercentage.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/uri/ShouldHaveAnchor.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/data/MapEntry.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ErrorHandler.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveFields.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/InstanceOfAssertFactories.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/IterableUtil.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/diff/DeleteDelta.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/StandardSoftAssertionsProvider.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/Strings.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractFloatArrayAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/exception/PathsException.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/ObjectArrayElementComparisonStrategy.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveSameSizeAs.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/ArrayWrapperList.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AssertDelegateTarget.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/OptionalShouldBePresent.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveSameDimensionsAs.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractStringAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainOnlyKeys.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/diff/DiffUtils.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractTemporalAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/filter/Filters.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/DoublePredicateAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/CharArrays.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeSymbolicLink.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/ElementsSatisfyingConsumer.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Digests.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractLocalDateAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/recursive/comparison/FieldLocation.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveScale.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeInSameSecondWindow.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/ErrorMessage.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainSubsequenceOfCharSequence.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/NoElementsShouldMatch.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEqualIgnoringHours.java



\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveAtLeastOneElementOfType.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/StandardComparisonStrategy.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveCause.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainOneOrMoreWhitespaces.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/diff/myers/MyersDiff.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/diff/myers/PathNode.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/FloatComparator.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeBeforeOrEqualTo.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractOffsetTimeAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Java6JUnitBDDSoftAssertions.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/junit/jupiter/InjectSoftAssertions.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Throwables.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractAtomicFieldUpdaterAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeGreater.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBePeriod.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractDoubleArrayAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Dates.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldSatisfy.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractPeriodAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeInSameMinute.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/introspection/FieldSupport.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/DoubleArrays.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveSameTime.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/filter/NotInFilter.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHavePropertyOrField.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/future/ShouldNotBeCompletedExceptionally.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/AtomicReferenceArrayElementComparisonStrategy.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldAccept.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/groups/FieldsOrPropertiesExtractor.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/OffsetDateTimeAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Java6SoftAssertions.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveTime.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractByteArrayAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/SoftAssertions.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeInstance.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveNoExtension.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/BooleanArrays.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AtomicReferenceAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveNoSuppressedExceptions.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Characters.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotStartWith.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/DescribableComparator.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/OptionalShouldContainInstanceOf.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractBigIntegerAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ByteArrayAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeSubsetOf.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Char2DArrays.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/Files.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/diff/DeltaComparator.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/CanIgnoreReturnValue.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/IndexedObjectEnumerableAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/Lists.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Arrays2D.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEqualWithinOffset.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractPredicateLikeAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveSize.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/condition/AnyOf.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/HamcrestCondition.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/uri/ShouldHaveHost.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/TypeComparators.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AtomicLongFieldUpdaterAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeAfterOrEqualTo.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeIn.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/uri/ShouldBeEqualToWithSortedQueryParameters.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/InputStreams.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/LocalDateTimeAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/FailureMessages.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/description/LazyTextDescription.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeInSameMinuteWindow.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldEndWith.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/OptionalShouldContain.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/uri/ShouldHavePort.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeInstanceOfAny.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/OptionalShouldBeEmpty.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/OptionalIntAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/AnyElementShouldMatch.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ClassBasedNavigableListAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AssertionsForClassTypes.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/CheckReturnValue.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ZonedDateTimeAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Java6BDDSoftAssertions.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AtomicReferenceArrayAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AtomicBooleanAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/SoftAssertionsRule.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/data/TemporalUnitLessThanOffset.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/description/EmptyTextDescription.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Assumptions.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/LongArrays.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldEndWithPath.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractIterableSizeAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveSameContent.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Abstract2DArrayAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ConstructorInvoker.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/WritableAssertionInfo.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/presentation/Representation.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ElementsShouldNotHave.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEmpty.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveToString.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveDigest.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractOffsetDateTimeAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/data/TemporalUnitWithinOffset.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeAssignableFrom.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ElementsShouldBe.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainOnlyNulls.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Short2DArrayAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/JUnitSoftAssertions.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveSizeLessThanOrEqualTo.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Comparables.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEqualNormalizingPunctuationAndWhitespace.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Lists.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/FactoryBasedNavigableListAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeAfter.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Byte2DArrays.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldOnlyHaveElementsOfTypes.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ClassModifierShouldBe.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/AbstractComparableNumberComparator.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/recursive/comparison/ComparisonDifference.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/OptionalAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/ChronoZonedDateTimeByInstantComparator.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ByteAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractLocalDateTimeAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotContainAtIndex.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/SubarraysShouldHaveSameSize.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/SpliteratorAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractCharArrayAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/diff/InsertDelta.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AtomicIntegerAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractZonedDateTimeAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/recursive/comparison/FieldMessages.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeLess.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/UnambiguousRepresentation.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AssertProvider.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotContain.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainSubsequence.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/VisibleForTesting.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveDuration.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractObjectArrayAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Predicates.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/future/ShouldBeDone.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/BooleanAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/Throwables.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeInThePast.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeLowerCase.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotExist.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AutoCloseableSoftAssertionsProvider.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Failures.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainValue.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/condition/Negative.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/TypeHolder.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEqualIgnoringSeconds.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/TypeMessages.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeAtIndex.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/diff/ChangeDelta.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/MultipleAssertionsError.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/BinaryDiff.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/future/ShouldBeCompletedWithin.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractLongArrayAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/uri/ShouldHaveParameter.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEqualIgnoringTimezone.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/IteratorAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ThrowableAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/IntegerAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Floats.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/future/ShouldBeCompleted.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractFileSizeAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotBe.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/junit/jupiter/SoftlyExtension.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/BigIntegers.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractLocalTimeAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/CollectionAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractMapSizeAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/IterableDiff.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/presentation/HexadecimalRepresentation.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/introspection/ClassUtils.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/MapAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotBeEmpty.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ShortAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/diff/Chunk.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/annotations/Beta.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ClassLoadingStrategyFactory.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHavePackage.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/future/Warning.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveValue.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/Sets.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeInSameSecond.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveMethods.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/SoftAssertionsProvider.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/DoubleArrayAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Long2DArrayAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ThrowableAssertAlternative.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ThrowableTypeAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldMatch.java



\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractAtomicReferenceAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ComparableAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/AssertJMultipleFailuresError.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeFinite.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeMixedCase.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ComparatorFactory.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractSoftAssertions.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AssertJProxySetup.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractUriAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/condition/Not.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEqualIgnoringMinutes.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Java6JUnitSoftAssertions.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/future/ShouldHaveFailed.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/PathAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveSuperclass.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHave.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AfterAssertionErrorCollected.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/GroupTypeDescription.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveReference.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/diff/myers/DiffNode.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/Maps.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeWritable.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AutoCloseableBDDSoftAssertions.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/MessageFormatter.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEqualIgnoringWhitespace.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveMessageMatchingRegex.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEmptyDirectory.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotContainValue.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/AssertionErrorMessagesAggregator.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/IntPredicateAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/data/TemporalUnitOffset.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AtomicReferenceFieldUpdaterAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeAfterYear.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Descriptable.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ThrowingConsumer.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldStartWith.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/extractor/ByNameSingleExtractor.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/presentation/BinaryRepresentation.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveStamp.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Java6BDDSoftAssertionsProvider.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractComparableAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveSizeGreaterThanOrEqualTo.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveExactlyTypes.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/diff/Delta.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/CompletableFutureAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Int2DArrayAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/data/Index.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/ByteArrays.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/presentation/CompositeRepresentation.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/ExtendedByTypesComparator.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Char2DArrayAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AutoCloseableSoftAssertions.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotBeEqual.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/ShortArrays.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/configuration/Services.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainEntries.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotBeEqualWithinPercentage.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Java6BDDAssertions.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/DeepDifference.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotContainPattern.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/PeriodAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ElementsShouldMatch.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeExactlyInstanceOf.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeExhausted.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/PredicateAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Double2DArrays.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractShortArrayAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/FloatArrayAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotBeNull.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotMatchPattern.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeTrue.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/future/ShouldNotHaveFailed.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/Preconditions.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/OffsetTimeAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/iterable/Extractor.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainNull.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/IntArrays.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AssertFactory.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AssumptionExceptionFactory.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotAccept.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractOptionalLongAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldMatchPattern.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveNoFields.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/BigIntegerComparator.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Arrays.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/InstanceOfAssertFactory.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/FloatingPointNumberAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/presentation/NumberGrouping.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Condition.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotBeEqualNormalizingWhitespace.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeFile.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ConditionAndGroupGenericTypeShouldBeTheSame.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotBeEqualIgnoringWhitespace.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeInSameYear.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/diff/myers/Equalizer.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/groups/Tuple.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotBeBlank.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeSorted.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AtomicLongAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/FieldByFieldComparator.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/RealNumbers.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveLineCount.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeAbsolutePath.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/NaturalOrderComparator.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEqualByComparingFieldByFieldRecursively.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBe.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/ConfigurableRecursiveFieldByFieldComparator.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/DoubleComparator.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Paths.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Longs.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/filter/NotFilter.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainCharSequenceOnlyOnce.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/FileSizeAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/SoftAssertionError.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractOptionalDoubleAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/recursive/comparison/FieldHolder.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeOfClassIn.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeBase64.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/condition/VerboseCondition.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Object2DArrays.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveSizeLessThan.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ElementsShouldBeAtMost.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainSequence.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/BDDSoftAssertionsProvider.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ObjectArrayAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/BigDecimals.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainValues.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/TriFunction.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/future/ShouldNotBeCancelled.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Diff.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Shorts.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ExtensionPoints.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/data/TemporalOffset.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotBeExactlyInstanceOf.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/OptionalDoubleShouldHaveValueCloseToOffset.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/OnFieldsComparator.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotBeBetween.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/uri/ShouldHaveFragment.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/FactoryBasedNavigableIterableAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/DateAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/introspection/MemberUtils.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveSizeGreaterThan.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/Streams.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/NullSafeComparator.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/LongArrayAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Conditions.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/StringAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeAnnotation.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/LongAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/DoubleAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveContent.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotBeIn.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Spliterators.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEqualByComparingOnlyGivenFields.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/PathNaturalOrderComparator.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/Hexadecimals.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Futures.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainRecursively.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotContainKeys.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeGreaterOrEqual.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/future/ShouldNotBeCompleted.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/InputStreamsException.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/introspection/IntrospectionError.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Java6StandardSoftAssertionsProvider.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/recursive/comparison/DualValue.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AtomicIntegerFieldUpdaterAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/CommonErrors.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/JUnitJupiterBDDSoftAssertions.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ElementsShouldBeAtLeast.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainEntry.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/FileAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AtomicStampedReferenceAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeLessOrEqual.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeNullOrEmpty.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveCauseInstance.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Assert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotHaveThrown.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEqual.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/groups/Properties.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/UrlAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainOnlyOnce.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AtomicMarkableReferenceAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Classes.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/InstantAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEqualIgnoringNanos.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/xml/XmlStringPrettyFormatter.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/configuration/PreferredAssumptionException.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotContainNull.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/data/Percentage.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/extractor/ResultOfExtractor.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Iterators.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/WithAssumptions.java



\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/AssertionErrorFactory.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/LongPredicateAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ElementsShouldNotBe.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotContainSubsequence.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEqualNormalizingWhitespace.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainOnlyWhitespaces.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/JUnitJupiterSoftAssertions.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotContainOnlyWhitespaces.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/BigDecimalComparator.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/configuration/ConfigurationProvider.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractFutureAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/CharSequenceAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveMessage.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ProxifyMethodChangingTheObjectUnderTest.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/uri/ShouldHaveScheme.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Bytes.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractPathAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/Objects.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ElementsShouldHave.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/Object2DArrayAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeDirectory.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeInSameDay.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Numbers.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveSizeBetween.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/uri/ShouldHaveAuthority.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/introspection/PropertyOrFieldSupport.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/ObjectArrays.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainAtIndex.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveDateField.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeFalse.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeExecutable.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotBeSame.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/future/ShouldNotBeDone.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeBetween.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/NumberAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveNoCause.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotHaveAnyElementsOfTypes.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/diff/Patch.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractOptionalIntAssert.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotBeEqualIgnoringCase.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Strings.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/filter/FilterOperator.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainKey.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/uri/ShouldHaveUserInfo.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/extractor/ToStringExtractor.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/IgnoringFieldsComparator.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotBeEqualWithinOffset.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveAtIndex.java

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/presentation/PredicateDescription.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/extractor/ByNameMultipleExtractor.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotHaveSameClass.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldContainOnly.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldNotBeOfClassIn.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/DescriptionFormatter.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/Integers.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldBeEqualIgnoringCase.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/FloatArrays.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/ObjectAssertFactory.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/internal/ComparisonStrategy.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldHaveNonNullFields.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/util/URLs.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/uri/ShouldHaveProtocol.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ShouldStartWithPath.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/RecursiveComparisonAssert.java  
\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/error/ElementsShouldHaveAtMost.java

No license file was found, but licenses were detected in source scan.

/\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

\* the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

\* specific language governing permissions and limitations under the License.

\*

```
* Copyright 2012-2021 the original author or authors.
*/
/**
 * Base class for all assertions.
 *
 * @param <SELF> the "self" type of this assertion class. Please read "<a href="http://bit.ly/1IZIRcY"
 * target="_blank">Emulating 'self types' using Java Generics to simplify fluent API
 * implementation"
 * for more details.
 * @param <ACTUAL> the type of the "actual" value.
 *
 * @author Alex Ruiz
 * @author Joel Costigliola
 * @author Mikhail Mazursky
 * @author Nicolas François
 */
```

Found in path(s):

```
* /opt/cola/permits/1341130270_1654817020.9922328/0/assertj-core-3-22-0-sources-
jar/org/assertj/core/api/AbstractAssert.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance
 * with
 * the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
 * an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the
 * specific language governing permissions and limitations under the License.
 *
 * Copyright 2012-2021 the original author or authors.
 */
```

```
/**
 * Base class for all implementations of assertions for { @link Boolean}s.
 *
 * @param <SELF> the "self" type of this assertion class. Please read "<a href="http://bit.ly/1IZIRcY"
 * target="_blank">Emulating 'self types' using Java Generics to simplify fluent API
 * implementation"
 * for more details.
 *
 * @author Alex Ruiz
 * @author Yvonne Wang
 * @author David DIDIER
 * @author Ansgar Konermann
 */
```

\* @author Mikhail Mazursky

\*/

Found in path(s):

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractBooleanAssert.java

No license file was found, but licenses were detected in source scan.

/\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

\* the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

\* specific language governing permissions and limitations under the License.

\*

\* Copyright 2012-2021 the original author or authors.

\*/

/\*\*

\* Base class for all implementations of assertions for { @link Object}s.

\*

\* @param <SELF> the "self" type of this assertion class. Please read ["http://bit.ly/1IZIRcY"](http://bit.ly/1IZIRcY)

\* [target="\\_blank">Emulating 'self types' using Java Generics to simplify fluent API implementation</a>](#)

\* [for more details.](#)

\* @param <ACTUAL> the type of the "actual" value.

\*

\* @author Yvonne Wang

\* @author Alex Ruiz

\* @author Nicolas François

\* @author Mikhail Mazursky

\* @author Joel Costigliola

\* @author Libor Ondrusek

\*/

Found in path(s):

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractObjectAssert.java

No license file was found, but licenses were detected in source scan.

/\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

\* the License. You may obtain a copy of the License at

```
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the
* specific language governing permissions and limitations under the License.
*
* Copyright 2012-2021 the original author or authors.
*/
/**
* Assertions methods applicable to groups of objects (e.g. arrays or collections.)
*
* @param <SELF> the "self" type of this assertion class. Please read "<a href="http://bit.ly/1IZIRcY"
* target="_blank">Emulating 'self types' using Java Generics to simplify fluent API
implementation"
* for more details.
* @param <ELEMENT> the type of elements of the "actual" value.
*
* @author Yvonne Wang
* @author Alex Ruiz
* @author Nicolas François
* @author Mikhail Mazursky
* @author Joel Costigliola
* @author Nicolas François
* @author Florent Biville
*/
```

Found in path(s):

```
* /opt/cola/permits/1341130270_1654817020.9922328/0/assertj-core-3-22-0-sources-
jar/org/assertj/core/api/ObjectEnumerableAssert.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance
with
* the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the
* specific language governing permissions and limitations under the License.
*
* Copyright 2012-2021 the original author or authors.
*/
/**
```

\* Base class for all implementations of assertions for { @link Date}s.

\* <p>

\* Note that assertions with date parameter comes with two flavor, one is obviously a { @link Date} and the other is a { @link String} representing a Date.<br>

\* For the latter, the default format follows ISO 8901 : "yyyy-MM-dd", user can override it with a custom format by calling { @link #withDateFormat(DateFormat)}.<br>

\* The user custom format will then be used for all next Date assertions (i.e not limited to the current assertion) in the test suite.<br>

\* To turn back to default format, simply call { @link #withDefaultDateFormatsOnly()}.

\*

\* @param <SELF> the "self" type of this assertion class. Please read "<a href="http://bit.ly/1IZIRcY" target="\_blank">Emulating 'self types' using Java Generics to simplify fluent API implementation</a>" for more details.

\* @author Tomasz Nurkiewicz (thanks for giving assertions idea)

\* @author Joel Costigliola

\* @author Mikhail Mazursky

\* @author William Delanoue

\* @author Michal Kordas

\* @author Eddú Meléndez

\*/

Found in path(s):

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractDateAssert.java

No license file was found, but licenses were detected in source scan.

/\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

\* the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

\* specific language governing permissions and limitations under the License.

\*

\* Copyright 2012-2021 the original author or authors.

\*/

/\*\*

\* Base class for all implementations of assertions for { @link Integer}s.

\*

\* @param <SELF> the "self" type of this assertion class. Please read "<a href="http://bit.ly/1IZIRcY" target="\_blank">Emulating 'self types' using Java Generics to simplify fluent API

implementation</a>&quot;

\* for more details.

\*

\* @author Drummond Dawson  
\* @author Yvonne Wang  
\* @author David DIDIER  
\* @author Ansgar Konermann  
\* @author Alex Ruiz  
\* @author Joel Costigliola  
\* @author Mikhail Mazursky  
\* @author Nicolas François  
\* @author Cal027  
\*/

Found in path(s):

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractIntegerAssert.java

No license file was found, but licenses were detected in source scan.

/\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

\* the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on

\* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

\* specific language governing permissions and limitations under the License.

\*

\* Copyright 2012-2021 the original author or authors.

\*/

/\*\*

\* Base class for all implementations of assertions for { @link Map}s.

\*

\* @param <SELF> the "self" type of this assertion class. Please read &quot;<a href="http://bit.ly/1IZIRcY" target="\_blank">Emulating 'self types' using Java Generics to simplify fluent API

implementation</a>&quot;

\* for more details.

\* @param <ACTUAL> the type of the "actual" value.

\* @param <K> the type of keys in the map.

\* @param <V> the type of values in the map.

\*

\* @author David DIDIER

\* @author Yvonne Wang

\* @author Alex Ruiz

\* @author Mikhail Mazursky

\* @author Nicolas François

\* @author dorzey

\* @author Filip Hrisafov



\*/

Found in path(s):

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractMapAssert.java

No license file was found, but licenses were detected in source scan.

/\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

\* the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on

\* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

\* specific language governing permissions and limitations under the License.

\*

\* Copyright 2012-2021 the original author or authors.

\*/

/\*\*

\* Base class for all implementations of assertions for { @link Collection}s.

\* @param <SELF> the "self" type of this assertion class. Please read ["http://bit.ly/1IZIRcY"](http://bit.ly/1IZIRcY)

\* [target="\\_blank">Emulating 'self types' using Java Generics to simplify fluent API implementation</a>](#)

\* [for more details.](#)

\* @param <ACTUAL> the type of the "actual" value.

\* @param <ELEMENT> the type of elements of the "actual" value.

\* @param <ELEMENT\_ASSERT> used for navigational assertions to return the right assert type.

\*

\* @since 3.21.0

\*/

Found in path(s):

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractCollectionAssert.java

No license file was found, but licenses were detected in source scan.

/\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

\* the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on

\* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

\* specific language governing permissions and limitations under the License.

\*

\* Copyright 2012-2021 the original author or authors.

\*/

/\*\*

\* Base class for all implementations of assertions for { @link Throwable}s.

\*

\* @param <SELF> the "self" type of this assertion class. Please read ["http://bit.ly/1IZIRcY"](http://bit.ly/1IZIRcY)

\* target="\_blank">Emulating 'self types' using Java Generics to simplify fluent API

implementation</a>&quot;

\* for more details.

\* @param <ACTUAL> the type of the "actual" value.

\*

\* @author David DIDIER

\* @author Alex Ruiz

\* @author Joel Costigliola

\* @author Mikhail Mazursky

\* @author Jack Gough

\* @author Mike Gilchrist

\*/

Found in path(s):

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-

jar/org/assertj/core/api/AbstractThrowableAssert.java

No license file was found, but licenses were detected in source scan.

/\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

\* the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on

\* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

\* specific language governing permissions and limitations under the License.

\*

\* Copyright 2012-2021 the original author or authors.

\*/

/\*\*

\* Base class for all implementations of assertions for { @link Class}es.

\*

\* @param <SELF> the "self" type of this assertion class. Please read ["http://bit.ly/1IZIRcY"](http://bit.ly/1IZIRcY)

\* target="\_blank">Emulating 'self types' using Java Generics to simplify fluent API

implementation</a>&quot;

\* for more details.

\*  
\* @author William Delanoue  
\* @author Mikhail Mazursky  
\*/

Found in path(s):

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractClassAssert.java

No license file was found, but licenses were detected in source scan.

/\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

\* the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

\* specific language governing permissions and limitations under the License.

\*

\* Copyright 2012-2021 the original author or authors.

\*/

/\*\*

\* Base class for implementations of `{ @link ObjectEnumerableAssert }` whose actual value type is `{ @link Collection }`.

\*

\* @param <SELF> the "self" type of this assertion class. Please read ["http://bit.ly/1IZIRcY"](http://bit.ly/1IZIRcY)

\* `target="_blank">Emulating 'self types' using Java Generics to simplify fluent API`

implementation</a>"

\* `for more details.`

\* @param <ACTUAL> the type of the "actual" value.

\* @param <ELEMENT> the type of elements of the "actual" value.

\* @param <ELEMENT\_ASSERT> used for navigational assertions to return the right assert type.

\*

\* @author Yvonne Wang

\* @author Alex Ruiz

\* @author Mathieu Baechler

\* @author Joel Costigliola

\* @author Maciej Jaskowski

\* @author Nicolas François

\* @author Mikhail Mazursky

\* @author Mateusz Haligowski

\* @author Lovro Pandzic

\* @author Marko Bekhta

\*/

Found in path(s):

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractIterableAssert.java

No license file was found, but licenses were detected in source scan.

/\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

\* the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

\* specific language governing permissions and limitations under the License.

\*

\* Copyright 2012-2021 the original author or authors.

\*/

/\*\*

\* Base class for all implementations of assertions for { @link BigDecimal}s.

\*

\* @param <SELF> the "self" type of this assertion class. Please read &quot;<a href="http://bit.ly/1IZIRcY" target="\_blank">Emulating 'self types' using Java Generics to simplify fluent API

implementation</a>&quot;

\* for more details.

\*

\* @author Drummond Dawson

\* @author David DIDIER

\* @author Ted M. Young

\* @author Yvonne Wang

\* @author Alex Ruiz

\* @author Joel Costigliola

\* @author Mikhail Mazursky

\*/

Found in path(s):

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractBigDecimalAssert.java

No license file was found, but licenses were detected in source scan.

/\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

\* the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on  
\* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the  
\* specific language governing permissions and limitations under the License.

\*  
\* Copyright 2012-2021 the original author or authors.

\*/  
/\*\*

\* Base class for all implementations of assertions for { @link Character}s.

\*  
\* @param <SELF> the "self" type of this assertion class. Please read &quot;<a href="http://bit.ly/1IZIRcY"  
\* target="\_blank">Emulating 'self types' using Java Generics to simplify fluent API  
implementation</a>&quot;  
\* for more details.

\*  
\* @author Yvonne Wang  
\* @author David DIDIER  
\* @author Ansgar Konermann  
\* @author Alex Ruiz  
\* @author Joel Costigliola  
\* @author Mikhail Mazursky  
\*/

Found in path(s):

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-  
jar/org/assertj/core/api/AbstractCharacterAssert.java

No license file was found, but licenses were detected in source scan.

/\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance  
with  
\* the License. You may obtain a copy of the License at

\*  
\* <http://www.apache.org/licenses/LICENSE-2.0>

\*  
\* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on  
\* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the

\* specific language governing permissions and limitations under the License.

\*  
\* Copyright 2012-2021 the original author or authors.

\*/  
/\*\*

\* Base class for all implementations of assertions for { @link Long}s.

\*  
\* @param <SELF> the "self" type of this assertion class. Please read &quot;<a href="http://bit.ly/1IZIRcY"  
\* target="\_blank">Emulating 'self types' using Java Generics to simplify fluent API  
implementation</a>&quot;

\* for more details.  
\*  
\* @author Drummond Dawson  
\* @author Yvonne Wang  
\* @author David DIDIER  
\* @author Ansgar Konermann  
\* @author Alex Ruiz  
\* @author Joel Costigliola  
\* @author Mikhail Mazursky  
\* @author Nicolas François  
\* @author Cal027  
\*/

Found in path(s):

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractLongAssert.java

No license file was found, but licenses were detected in source scan.

/\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

\* the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

\* specific language governing permissions and limitations under the License.

\*

\* Copyright 2012-2021 the original author or authors.

\*/

/\*\*

\* Base class for all implementations of assertions for { @link LongAdder}s.

\*

\* @param <SELF> the "self" type of this assertion class. Please read ["http://bit.ly/1IZIRcY"](http://bit.ly/1IZIRcY)

\* [target="\\_blank">Emulating 'self types' using Java Generics to simplify fluent API implementation</a>](#)

\* for more details.

\*

\* @author Grzegorz Piwowarek

\* @since 3.16.0

\*/

Found in path(s):

\* /opt/cola/permits/1341130270\_1654817020.9922328/0/assertj-core-3-22-0-sources-jar/org/assertj/core/api/AbstractLongAdderAssert.java

# 1.76 jetbrains-kotlin-kotlin-stdlib-jdk8 1.7.20

## 1.76.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2010-2017 JetBrains s.r.o.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1526005728_1673461695.146515/0/kotlin-stdlib-jdk8-1-7-20-sources-
jar/kotlin/collections/Collections.kt
* /opt/cola/permits/1526005728_1673461695.146515/0/kotlin-stdlib-jdk8-1-7-20-sources-
jar/kotlin/text/RegexExtensions.kt
* /opt/cola/permits/1526005728_1673461695.146515/0/kotlin-stdlib-jdk8-1-7-20-sources-
jar/kotlin/streams/Streams.kt
* /opt/cola/permits/1526005728_1673461695.146515/0/kotlin-stdlib-jdk8-1-7-20-sources-
jar/kotlin/internal/jdk8/JDK8PlatformImplementations.kt
```

# 1.77 aws-java-sdk-::-third-party-::-jackson- dataformat-cbor 2.17.101

## 1.77.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,



and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and

limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:  
Apache Commons Lang  
Copyright 2001-2020 The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (<https://www.apache.org/>).

## 1.78 jackson-core 2.14.0

### 1.78.1 Available under license :

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta ([tatu.saloranta@iki.fi](mailto:tatu.saloranta@iki.fi)), and has been in development since 2007. It is currently developed by a community of developers.

## Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0 To find the details that apply to this artifact see the accompanying LICENSE file.

## ## Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade

names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
  
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
  
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier



identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

## 1.79 re2j 1.3

### 1.79.1 Available under license :

No license file was found, but licenses were detected in source scan.

// Copyright 2010 Google Inc. All Rights Reserved.

Found in path(s):

- \* /opt/cola/permits/1168630218\_1622042000.35/0/re2j-1-3-sources-3-jar/com/google/re2j/Pattern.java
- \* /opt/cola/permits/1168630218\_1622042000.35/0/re2j-1-3-sources-3-jar/com/google/re2j/Matcher.java

No license file was found, but licenses were detected in source scan.

// Copyright 2010 The Go Authors. All rights reserved.

Found in path(s):

- \* /opt/cola/permits/1168630218\_1622042000.35/0/re2j-1-3-sources-3-jar/com/google/re2j/RE2.java
- \* /opt/cola/permits/1168630218\_1622042000.35/0/re2j-1-3-sources-3-jar/com/google/re2j/Unicode.java
- \* /opt/cola/permits/1168630218\_1622042000.35/0/re2j-1-3-sources-3-jar/com/google/re2j/Machine.java
- \* /opt/cola/permits/1168630218\_1622042000.35/0/re2j-1-3-sources-3-jar/com/google/re2j/Parser.java
- \* /opt/cola/permits/1168630218\_1622042000.35/0/re2j-1-3-sources-3-jar/com/google/re2j/PatternSyntaxException.java
- \* /opt/cola/permits/1168630218\_1622042000.35/0/re2j-1-3-sources-3-jar/com/google/re2j/Compiler.java
- \* /opt/cola/permits/1168630218\_1622042000.35/0/re2j-1-3-sources-3-jar/com/google/re2j/MachineInput.java
- \* /opt/cola/permits/1168630218\_1622042000.35/0/re2j-1-3-sources-3-jar/com/google/re2j/Prog.java
- \* /opt/cola/permits/1168630218\_1622042000.35/0/re2j-1-3-sources-3-jar/com/google/re2j/Inst.java
- \* /opt/cola/permits/1168630218\_1622042000.35/0/re2j-1-3-sources-3-jar/com/google/re2j/CharClass.java
- \* /opt/cola/permits/1168630218\_1622042000.35/0/re2j-1-3-sources-3-jar/com/google/re2j/Utils.java
- \* /opt/cola/permits/1168630218\_1622042000.35/0/re2j-1-3-sources-3-jar/com/google/re2j/Regexp.java

No license file was found, but licenses were detected in source scan.

// Copyright 2011 The Go Authors. All rights reserved.

Found in path(s):

\* /opt/cola/permits/1168630218\_1622042000.35/0/re2j-1-3-sources-3-jar/com/google/re2j/Simplify.java

# 1.80 byte-buddy byte-buddy-1.11.3

## 1.80.1 Available under license :

ASM: a very small and fast Java bytecode manipulation framework

Copyright (c) 2000-2011 INRIA, France Telecom

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product bundles ASM 9.1, which is available under a "3-clause BSD"

license. For details, see licenses/ASM. For more information visit <https://asm.ow2.io>.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. **Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS  
Copyright 2014 - `#{current.year}` Rafael Winterhalter

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## 1.81 slf4j-api-module 1.7.36

### 1.81.1 Available under license :

No license file was found, but licenses were detected in source scan.

/\*\*

\* Copyright (c) 2004-2011 QOS.ch

\* All rights reserved.

\*

\* Permission is hereby granted, free of charge, to any person obtaining

\* a copy of this software and associated documentation files (the

\* "Software"), to deal in the Software without restriction, including  
 \* without limitation the rights to use, copy, modify, merge, publish,  
 \* distribute, sublicense, and/or sell copies of the Software, and to  
 \* permit persons to whom the Software is furnished to do so, subject to  
 \* the following conditions:  
 \*  
 \* The above copyright notice and this permission notice shall be  
 \* included in all copies or substantial portions of the Software.  
 \*  
 \* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,  
 \* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF  
 \* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND  
 \* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE  
 \* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION  
 \* OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION  
 \* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.  
 \*  
 \*/

Found in path(s):

\* /opt/cola/permits/1331474007\_1653510300.1759446/0/slf4j-api-1-7-36-sources-jar/org/slf4j/helpers/FormattingTuple.java  
 \* /opt/cola/permits/1331474007\_1653510300.1759446/0/slf4j-api-1-7-36-sources-jar/org/slf4j/helpers/NOPLoggerFactory.java  
 \* /opt/cola/permits/1331474007\_1653510300.1759446/0/slf4j-api-1-7-36-sources-jar/org/slf4j/spi/MDCAdapter.java  
 \* /opt/cola/permits/1331474007\_1653510300.1759446/0/slf4j-api-1-7-36-sources-jar/org/slf4j/MarkerFactory.java  
 \* /opt/cola/permits/1331474007\_1653510300.1759446/0/slf4j-api-1-7-36-sources-jar/org/slf4j/impl/StaticLoggerBinder.java  
 \* /opt/cola/permits/1331474007\_1653510300.1759446/0/slf4j-api-1-7-36-sources-jar/org/slf4j/impl/StaticMarkerBinder.java  
 \* /opt/cola/permits/1331474007\_1653510300.1759446/0/slf4j-api-1-7-36-sources-jar/org/slf4j/MDC.java  
 \* /opt/cola/permits/1331474007\_1653510300.1759446/0/slf4j-api-1-7-36-sources-jar/org/slf4j/helpers/MessageFormatter.java  
 \* /opt/cola/permits/1331474007\_1653510300.1759446/0/slf4j-api-1-7-36-sources-jar/org/slf4j/Marker.java  
 \* /opt/cola/permits/1331474007\_1653510300.1759446/0/slf4j-api-1-7-36-sources-jar/org/slf4j/helpers/BasicMDCAdapter.java  
 \* /opt/cola/permits/1331474007\_1653510300.1759446/0/slf4j-api-1-7-36-sources-jar/org/slf4j/helpers/BasicMarker.java  
 \* /opt/cola/permits/1331474007\_1653510300.1759446/0/slf4j-api-1-7-36-sources-jar/org/slf4j/ILoggerFactory.java  
 \* /opt/cola/permits/1331474007\_1653510300.1759446/0/slf4j-api-1-7-36-sources-jar/org/slf4j/impl/StaticMDCBinder.java  
 \* /opt/cola/permits/1331474007\_1653510300.1759446/0/slf4j-api-1-7-36-sources-jar/org/slf4j/helpers/MarkerIgnoringBase.java  
 \* /opt/cola/permits/1331474007\_1653510300.1759446/0/slf4j-api-1-7-36-sources-jar/org/slf4j/helpers/NOPMDCAdapter.java  
 \* /opt/cola/permits/1331474007\_1653510300.1759446/0/slf4j-api-1-7-36-sources-jar/org/slf4j/helpers/NamedLoggerBase.java

```

* /opt/cola/permits/1331474007_1653510300.1759446/0/slf4j-api-1-7-36-sources-
jar/org/slf4j/helpers/BasicMarkerFactory.java
* /opt/cola/permits/1331474007_1653510300.1759446/0/slf4j-api-1-7-36-sources-jar/org/slf4j/helpers/Util.java
* /opt/cola/permits/1331474007_1653510300.1759446/0/slf4j-api-1-7-36-sources-jar/org/slf4j/LoggerFactory.java
* /opt/cola/permits/1331474007_1653510300.1759446/0/slf4j-api-1-7-36-sources-
jar/org/slf4j/spi/LoggerFactoryBinder.java
* /opt/cola/permits/1331474007_1653510300.1759446/0/slf4j-api-1-7-36-sources-
jar/org/slf4j/spi/LocationAwareLogger.java
* /opt/cola/permits/1331474007_1653510300.1759446/0/slf4j-api-1-7-36-sources-
jar/org/slf4j/helpers/SubstituteLogger.java
* /opt/cola/permits/1331474007_1653510300.1759446/0/slf4j-api-1-7-36-sources-jar/org/slf4j/Logger.java
* /opt/cola/permits/1331474007_1653510300.1759446/0/slf4j-api-1-7-36-sources-
jar/org/slf4j/spi/MarkerFactoryBinder.java
* /opt/cola/permits/1331474007_1653510300.1759446/0/slf4j-api-1-7-36-sources-
jar/org/slf4j/helpers/NOPLogger.java
* /opt/cola/permits/1331474007_1653510300.1759446/0/slf4j-api-1-7-36-sources-jar/org/slf4j/IMarkerFactory.java
* /opt/cola/permits/1331474007_1653510300.1759446/0/slf4j-api-1-7-36-sources-
jar/org/slf4j/helpers/SubstituteLoggerFactory.java

```

# 1.82 kotlinox-coroutines-core 1.1.1

## 1.82.1 Available under license :

```

=====
== NOTICE file corresponding to the section 4 d of ==
== the Apache License, Version 2.0, ==
== in this case for the kotlinox.coroutines library. ==
=====

```

kotlinox.coroutines library.

Copyright 2016-2018 JetBrains s.r.o and respective authors and developers

Copyright 2016-2018 JetBrains s.r.o.

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

The MIT License (MIT)

Copyright (c) 2016 Parker Moore



Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.83 google-guava 31.0.1-jre

### 1.83.1 Available under license :

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2011 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

\* in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License

\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

\* or implied. See the License for the specific language governing permissions and limitations under

\* the License.

\*/

/\*

\* This method was written by Doug Lea with assistance from members of JCP JSR-166 Expert Group

\* and released to the public domain, as explained at

\* <http://creativecommons.org/licenses/publicdomain>

\*

\* As of 2010/06/11, this method is identical to the (package private) hash method in OpenJDK 7's

\* `java.util.HashMap` class.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/Striped.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2017 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except  
\* in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License  
\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express

\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/primitives/ImmutableIntArray.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/primitives/ImmutableLongArray.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/primitives/ImmutableDoubleArray.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/ForwardingCondition.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/hash/AbstractHashFunction.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/ForwardingLock.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2009 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except  
\* in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License  
\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express

\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.

\*/

```
/**
 * Outer class that exists solely to let us write {@code Partially.GwtIncompatible} instead of plain
 * {@code GwtIncompatible}. This is more accurate for {@link Futures#catching}, which is available
 * under GWT but with a slightly different signature.
 *
 * <p>We can't use {@code PartiallyGwtIncompatible} because then the GWT compiler wouldn't recognize
 * it as a {@code GwtIncompatible} annotation. And for {@code Futures.catching}, we need the GWT
 * compiler to autostrip the normal server method in order to expose the special, inherited GWT
 * version.
 */
```

Found in path(s):

```
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
jar/com/google/common/util/concurrent/Partially.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2007 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

```
/**
 * Returns an array containing all of the elements in the specified collection. This method
 * returns the elements in the order they are returned by the collection's iterator. The returned
 * array is "safe" in that no references to it are maintained by the collection. The caller is
 * thus free to modify the returned array.
 *
 * <p>This method assumes that the collection size doesn't change while the method is running.
 *
 * <p>TODO(kevinb): support concurrently modified collections?
 *
 * @param c the collection for which to return an array of elements
 */
```

Found in path(s):

```
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
jar/com/google/common/collect/ObjectArrays.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2008 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
/*
 * This method was rewritten in Java from an intermediate step of the Murmur hash function in
 * http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
 * following header:
 *
 * MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
 * hereby disclaims copyright to this source code.
 */
```

Found in path(s):

```
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
jar/com/google/common/collect/Hashing.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2010 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
```

jar/com/google/common/base/Strings.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/SortedLists.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/annotations/Beta.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/UncaughtExceptionHandler.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/primitives/package-info.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/Atomics.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/Monitor.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/ListeningExecutorService.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/ThreadFactoryBuilder.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/annotations/package-info.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/ContiguousSet.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/Ascii.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/Equivalence.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/ForwardingBlockingQueue.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/net/package-info.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2009 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except  
\* in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the

\* License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,  
either

\* express or implied. See the License for the specific language governing permissions and

\* limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-

jar/com/google/common/collect/ImmutableSortedAsList.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2007 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

\* in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License

\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express

\* or implied. See the License for the specific language governing permissions and limitations under

\* the License.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/base/Suppliers.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/Interners.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/Files.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/primitives/Primitives.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/DirectExecutor.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/base/Function.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/LineBuffer.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/MultiInputStream.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/package-info.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/base/Predicate.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/eventbus/EventBus.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/LineReader.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/eventbus/AsyncEventBus.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/base/AbstractIterator.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-

jar/com/google/common/eventbus/Subscribe.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/eventbus/AllowConcurrentEvents.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/io/CountingInputStream.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/Supplier.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/io/CharStreams.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/io/Flushables.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/FinalizableReferenceQueue.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/Throwables.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/Preconditions.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/AbstractFuture.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/io/Closeables.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/Charsets.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/ListenableFuture.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/io/Resources.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/io/LittleEndianDataOutputStream.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/Functions.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/EnumMultiset.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/FinalizableWeakReference.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/Defaults.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/FinalizableSoftReference.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/ExecutionList.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/io/CountingOutputStream.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/FinalizablePhantomReference.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/io/LittleEndianDataInputStream.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-

jar/com/google/common/util/concurrent/package-info.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/FinalizableReference.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/Objects.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/eventbus/DeadEvent.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/package-info.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/eventbus/package-info.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/io/ByteStreams.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/HashBiMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/Predicates.java  
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/HashMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/NullsFirstOrdering.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/TreeMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/Ordering.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/AbstractMapBasedMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/EnumBiMap.java



\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/Maps.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/AbstractBiMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ForwardingSortedMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/AbstractMapBasedMultiset.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ExplicitOrdering.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ForwardingMapEntry.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/NullsLastOrdering.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/RegularImmutableSet.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/SetMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/AbstractListMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/Iterables.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/HashMultiset.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/LinkedHashMultiset.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/Multisets.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ForwardingListIterator.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ForwardingMultiset.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/NaturalOrdering.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ClassToInstanceMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ReverseNaturalOrdering.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/SingletonImmutableSet.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/AbstractSortedSetMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/SortedSetMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/TreeMultiset.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/LexicographicalOrdering.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ForwardingSortedSet.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ForwardingMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/Sets.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/UsingToStringOrdering.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/AbstractIterator.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ReverseOrdering.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ForwardingObject.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableSet.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/MapDifference.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableList.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/Iterators.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ArrayListMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ForwardingList.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ForwardingQueue.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/LinkedHashMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/Multimaps.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/AbstractMultiset.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/MutableClassToInstanceMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ForwardingIterator.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ComparatorOrdering.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/Synchronized.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/EnumHashBiMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/package-info.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/AbstractSetMultimap.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/Interner.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ConcurrentHashMultiset.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ByFunctionOrdering.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/CompoundOrdering.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ForwardingConcurrentMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/LinkedListMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ForwardingSet.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/Lists.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/BiMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ForwardingCollection.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/AbstractMapEntry.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ListMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ForwardingMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/Multiset.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/Multimap.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright (C) 2011 The Guava Authors  
\*  
\* Licensed under the Apache License, Version 2.0 (the "License");  
\* you may not use this file except in compliance with the License.  
\* You may obtain a copy of the License at  
\*  
\* <http://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS,  
\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
\* See the License for the specific language governing permissions and  
\* limitations under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/AtomicLongMap.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/GwtTransient.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2009 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

/\*\*

\* Not supported. <b>You are attempting to create a map that may contain a non-{@code Comparable}

\* key.</b> Proper calls will resolve to the version in {@code ImmutableSortedMap}, not this dummy

\* version.

\*

\* @throws UnsupportedOperationException always

\* @deprecated <b>Pass a key of type {@code Comparable} to use {@link

\* `ImmutableSortedMap#of(Comparable, Object)`.</b>

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableSortedMapFauxverideShim.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2012 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

\* in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License

\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

express

\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.

\*/

/\*

\* This method was rewritten in Java from an intermediate step of the Murmur hash function in  
\* <http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp>, which contained the  
\* following header:

\*

\* MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author  
\* hereby disclaims copyright to this source code.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/SmallCharMatcher.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2011 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except  
\* in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the

\* License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,  
either

\* express or implied. See the License for the specific language governing permissions and

\* limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/ImmutableSortedMultiset.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/RegularImmutableSortedMultiset.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/GeneralRange.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/ImmutableSortedMultisetFauxverideShim.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/SortedIterable.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/SortedIterables.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/RangeSet.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-

jar/com/google/common/collect/ForwardingSortedMultiset.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/AbstractRangeSet.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/Count.java  
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2018 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/hash/ImmutableSupplier.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/JdkBackedImmutableSet.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/ExecutionSequencer.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/JdkBackedImmutableMultiset.java  
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/math/BigDecimalMath.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/Java8Compatibility.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/OverflowAvoidingLockSupport.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/base/Java8Compatibility.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/hash/Java8Compatibility.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/math/ToDoubleRounder.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2005 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except  
\* in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License  
\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
\* express

\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/reflect/Reflection.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2013 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except  
\* in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License  
\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
\* express

\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/base/VerifyException.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/reflect/TypeVisitor.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/base/Utf8.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/WrappingScheduledExecutorService.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/AbstractTable.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/CharSequenceReader.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/FilteredMultimapValues.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/eventbus/SubscriberExceptionHandler.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/thirdparty/publicsuffix/PublicSuffixType.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/eventbus/SubscriberExceptionContext.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/Runnables.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/base/Verify.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/hash/HashingInputStream.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2020 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except  
\* in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License  
\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
\* express

\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.

\*/

/\*\*

\* Holder for web specializations of methods of { @code Doubles}. Intended to be empty for regular  
\* version.



\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/primitives/DoublesMethodsForWeb.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2011 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

\* in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License

\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

\* or implied. See the License for the specific language governing permissions and limitations under

\* the License.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/math/LongMath.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/cache/ForwardingCache.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/ExecutionError.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/hash/AbstractHasher.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/EmptyContiguousSet.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/hash/HashCode.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/hash/AbstractCompositeHashFunction.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/hash/Hashing.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/WrappingExecutorService.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/base/Present.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/Uninterruptibles.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/net/HostAndPort.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-

jar/com/google/common/reflect/Types.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/FutureCallback.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/math/DoubleMath.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/PairwiseEquivalence.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/hash/HashingOutputStream.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/hash/AbstractNonStreamingHashFunction.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/hash/Hasher.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/DescendingImmutableSortedMultiset.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/AsyncFunction.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/cache/CacheBuilderSpec.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/math/IntMath.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/cache/CacheStats.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/CycleDetectingLockFactory.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/RegularContiguousSet.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/AbstractSortedMultiset.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/hash/MessageDigestHashFunction.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/math/BigIntegerMath.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/primitives/UnsignedInts.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/math/package-info.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/BoundType.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/math/MathPreconditions.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/ForwardingExecutorService.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/hash/Funnels.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/Queues.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-

jar/com/google/common/base/Ticker.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/TreeRangeSet.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/reflect/TypeParameter.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/base/FunctionalEquivalence.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/hash/HashFunction.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/primitives/UnsignedLongs.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/cache/RemovalListeners.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/cache/AbstractCache.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/RegularImmutableMultiset.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/hash/Murmur3\_128HashFunction.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/hash/AbstractStreamingHasher.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/cache/AbstractLoadingCache.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/cache/Cache.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/hash/Crc32cHashFunction.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/cache/RemovalCause.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/hash/Funnel.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/primitives/UnsignedLong.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/hash/BloomFilterStrategies.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/base/Optional.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/cache/LoadingCache.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/hash/PrimitiveSink.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/primitives/ParseRequest.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/ListeningScheduledExecutorService.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/UncheckedExecutionException.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-

jar/com/google/common/util/concurrent/AbstractScheduledService.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/hash/BloomFilter.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/math/DoubleUtils.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/AbstractListeningExecutorService.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/cache/Weigher.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/hash/Murmur3\_32HashFunction.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/ForwardingListeningExecutorService.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/cache/RemovalNotification.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/Enums.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/cache/ForwardingLoadingCache.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/net/MediaType.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/primitives/UnsignedInteger.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/cache/RemovalListener.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/Absent.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/cache/package-info.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/net/HttpHeaders.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/cache/CacheLoader.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2016 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/AbstractUndirectedNetworkConnections.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/MutableValueGraph.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/ImmutableValueGraph.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/DirectedNetworkConnections.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/AbstractDirectedNetworkConnections.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/StandardNetwork.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ArrayListMultimapGwtSerializationDependencies.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/HashMultimapGwtSerializationDependencies.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/GraphConnections.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/StandardMutableValueGraph.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/AbstractNetwork.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/UndirectedGraphConnections.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableMultisetGwtSerializationDependencies.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/AbstractGraphBuilder.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/UndirectedMultiNetworkConnections.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/ElementOrder.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/MultiEdgesConnecting.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/StandardMutableNetwork.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/EdgesConnecting.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/StandardMutableGraph.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/StandardValueGraph.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/EndpointPairIterator.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/UndirectedNetworkConnections.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/MoreCollectors.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/ForwardingNetwork.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/DirectedGraphConnections.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/CollectCollectors.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/GraphBuilder.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/MapIteratorCache.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/Comparators.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/ForwardingGraph.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/ValueGraphBuilder.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/DirectedMultiNetworkConnections.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/EndpointPair.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/RangeGwtSerializationDependencies.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/AbstractValueGraph.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/LinkedHashMultimapGwtSerializationDependencies.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/NetworkConnections.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/GraphConstants.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/ValueGraph.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/MapRetrievalCache.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/NetworkBuilder.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/AbstractGraph.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/ForwardingValueGraph.java  
No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2006 The Guava Authors

\*  
\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except  
\* in compliance with the License. You may obtain a copy of the License at  
\*  
\* <http://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software distributed under the License  
\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express  
\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/PatternFilenameFilter.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/GwtFluentFutureCatchingSpecialization.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/TimeLimiter.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/reflect/TypeToken.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/annotations/VisibleForTesting.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/FakeTimeLimiter.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/FuturesGetChecked.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/AppendableWriter.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/AbstractTransformFuture.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/escape/CharEscaper.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/AggregateFuture.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/TimeoutFuture.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/SimpleTimeLimiter.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/CollectionFuture.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/GwtFuturesCatchingSpecialization.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/FluentFuture.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/ImmediateFuture.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/AbstractCatchingFuture.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/escape/CharEscaperBuilder.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/UncheckedTimeoutException.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/base/CaseFormat.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/Futures.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2018 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/JdkBackedImmutableBiMap.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/BaseImmutableMultimap.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/IndexedImmutableSet.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/JdkBackedImmutableMap.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2019 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

\* in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License



\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express  
\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/primitives/Platform.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/Internal.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2019 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/CompactHashing.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/IncidentEdgeSet.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2020 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

\* in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License

\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

\* or implied. See the License for the specific language governing permissions and limitations under

\* the License.

\*/

/\*\*

\* Holder for web specializations of methods of { @code Ints }. Intended to be empty for regular  
\* version.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/primitives/IntsMethodsForWeb.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2021 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except  
\* in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License  
\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express

\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/NullnessCasts.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/NullnessCasts.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/NullnessCasts.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2011 The Guava Authors.

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except  
\* in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License  
\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express

\* or implied. See the License for the specific language governing permissions and limitations under

\* the License.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/hash/package-info.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2008 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except  
\* in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License  
\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express

\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/escape/UnicodeEscaper.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/base/CharMatcher.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/SequentialExecutor.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/base/Joiner.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/base/internal/Finalizer.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/FluentIterable.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/primitives/Ints.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/base/Stopwatch.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/net/PercentEscaper.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/primitives/Floats.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/primitives/Longs.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/thirdparty/publicsuffix/TrieParser.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/MultiReader.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/primitives/Bytes.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/primitives/Doubles.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/ListenableFutureTask.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/escape/Escaper.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/primitives/Chars.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/FileBackedOutputStream.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/primitives/Booleans.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/net/InetAddresses.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/primitives/Shorts.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/base/Converter.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2012 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/FilteredSetMultimap.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ForwardingImmutableList.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableEnumMap.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-

jar/com/google/common/collect/ForwardingImmutableSet.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/EvictingQueue.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/RegularImmutableAsList.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/ForwardingBlockingDeque.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/ForwardingNavigableMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/TransformedIterator.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/DescendingMultiset.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/SortedMultisetBridge.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/TreeTraverser.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/FilteredMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/FilteredKeySetMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/AbstractSortedKeySortedSetMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/TransformedListIterator.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/CompactLinkedHashSet.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/FilteredEntryMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/CompactHashMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/CompactLinkedHashMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/CompactHashSet.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/DescendingImmutableSortedSet.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/ForwardingBlockingDeque.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/FilteredKeyListMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/ForwardingImmutableMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/AbstractMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/ForwardingNavigableSet.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-

jar/com/google/common/collect/AbstractNavigableMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/AllEqualOrdering.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/ForwardingDeque.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/RangeMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/TreeRangeMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/FilteredEntrySetMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/UnmodifiableSortedMultiset.java  
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2010 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/SortedMapDifference.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/ForwardingSortedSetMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/ForwardingImmutableCollection.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/RowSortedTable.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/ForwardingListMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/AbstractSequentialIterator.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/ForwardingSetMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/UnmodifiableListIterator.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/MinMaxPriorityQueue.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2011 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/SortedMultisets.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/SortedMultiset.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2007 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

\*/

/\*

\* This following method is a modified version of one found in

\* <http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/test/tck/AbstractExecutorServiceTest.java?revision=1.30>

\* which contained the following notice:

\*

\* Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to

\* the public domain, as explained at <http://creativecommons.org/publicdomain/zero/1.0/>

\*

\* Other contributors include Andrew Wright, Jeffrey Hayes, Pat Fisher, Mike Judd.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/MoreExecutors.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2015 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you

\* may not use this file except in compliance with the License. You may

\* obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

\* implied. See the License for the specific language governing

\* permissions and limitations under the License.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/Streams.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2012 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

\* in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License

\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

\* or implied. See the License for the specific language governing permissions and limitations under

\* the License.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/math/LinearTransformation.java



\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/BaseEncoding.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/reflect/ClassPath.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/base/StandardSystemProperty.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/ListenableScheduledFuture.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/reflect/TypeCapture.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/cache/LongAddable.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/math/StatsAccumulator.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/html/package-info.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/SmoothRateLimiter.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/escape/package-info.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/reflect/package-info.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableRangeMap.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/math/PairedStats.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/ByteSink.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/FilteredKeyMultimap.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/reflect/MutableTypeToInstanceMap.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/Closer.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/hash/LongAddables.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/RateLimiter.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/math/PairedStatsAccumulator.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/CharSource.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/CharSink.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/reflect/TypeToInstanceMap.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/xml/package-info.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableRangeSet.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/hash/SipHashFunction.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/reflect/ImmutableTypeToInstanceMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/cache/LongAddables.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/hash/AbstractByteHasher.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/ByteSource.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/reflect/Invokable.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/hash/LongAddable.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/math/Stats.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/ServiceManager.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/reflect/Parameter.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/hash/ChecksumHashFunction.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/reflect/AbstractInvocationHandler.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/CartesianList.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/FileWriteMode.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright (C) 2009 The Guava Authors  
\*  
\* Licensed under the Apache License, Version 2.0 (the "License");  
\* you may not use this file except in compliance with the License.  
\* You may obtain a copy of the License at  
\*  
\* <http://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS,  
\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
\* See the License for the specific language governing permissions and  
\* limitations under the License.  
\*/

Found in path(s):

- \* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableSortedMap.java
- \* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/RegularImmutableList.java
- \* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/RegularImmutableSortedSet.java
- \* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableSetMultimap.java
- \* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/AbstractIndexedListIterator.java
- \* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableEnumSet.java
- \* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableAsList.java
- \* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/TableCollectors.java
- \* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableClassToInstanceMap.java
- \* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ArrayTable.java
- \* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/SingletonImmutableTable.java
- \* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/DiscreteDomain.java
- \* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ComparisonChain.java
- \* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/SingletonImmutableList.java
- \* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ForwardingTable.java
- \* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/EmptyImmutableSetMultimap.java
- \* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableSortedSetFauxverideShim.java
- \* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ComputationException.java
- \* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableTable.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2014 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS,  
\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
\* See the License for the specific language governing permissions and  
\* limitations under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/MutableGraph.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/SuccessorsFunction.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/InsecureRecursiveDeleteException.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/ImmutableNetwork.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/Network.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/Graph.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/RecursiveDeleteOption.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/MutableNetwork.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/Graphs.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/TopKSelector.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/PredecessorsFunction.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/ImmutableGraph.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* Written by Doug Lea with assistance from members of JCP JSR-166  
\* Expert Group and released to the public domain, as explained at  
\* <http://creativecommons.org/publicdomain/zero/1.0/>  
\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/AtomicDoubleArray.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/cache/Striped64.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-

jar/com/google/common/hash/Striped64.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/cache/LongAdder.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/hash/LongAdder.java  
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2014 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/ListenerCallQueue.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/TrustedListenableFutureTask.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/math/Quantiles.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/eventbus/SubscriberRegistry.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/MoreObjects.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/eventbus/Dispatcher.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/eventbus/Subscriber.java  
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2020 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
```

\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express  
\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.  
\*/  
/\*\*  
\* Holder for web specializations of methods of { @code Shorts}. Intended to be empty for regular  
\* version.  
\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/primitives/ShortsMethodsForWeb.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright (C) 2008 The Guava Authors  
\*  
\* Licensed under the Apache License, Version 2.0 (the "License");  
\* you may not use this file except in compliance with the License.  
\* You may obtain a copy of the License at  
\*  
\* <http://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS,  
\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
\* See the License for the specific language governing permissions and  
\* limitations under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/TreeBasedTable.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableMultiset.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/PeekingIterator.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/StandardRowSortedTable.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/Collections2.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableMapEntrySet.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/Platform.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/Table.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/RegularImmutableMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableSortedSet.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/CollectPreconditions.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/RegularImmutableBiMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/thirdparty/publicsuffix/PublicSuffixPatterns.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableMapKeySet.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableBiMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableMapValues.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/StandardTable.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableEntry.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableListMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/SingletonImmutableBiMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/HashBasedTable.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/EmptyImmutableListMultimap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/UnmodifiableIterator.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/Tables.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ImmutableCollection.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/Serialization.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/Range.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2016 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

\* in compliance with the License. You may obtain a copy of the License at  
\*  
\* <http://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software distributed under the License  
\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express  
\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/CommonPattern.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/PatternCompiler.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/CommonMatcher.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/JdkPattern.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2020 The Guava Authors

\*

\* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except  
\* in compliance with the License. You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software distributed under the License  
\* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express

\* or implied. See the License for the specific language governing permissions and limitations under  
\* the License.

\*/

/\*\*

\* Holder for web specializations of methods of { @code Floats }. Intended to be empty for regular  
\* version.

\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/primitives/FloatsMethodsForWeb.java

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (C) 2017 The Guava Authors



```
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
jar/com/google/common/util/concurrent/ClosingFuture.java
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
jar/com/google/common/graph/Traverser.java
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
jar/com/google/common/graph/BaseGraph.java
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
jar/com/google/common/graph/AbstractBaseGraph.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2016 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for extra methods of {@code Objects} only in web. Intended to be empty for regular
* version.
*/
```

Found in path(s):

```
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
jar/com/google/common/base/ExtraObjectsMethodsForWeb.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2015 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
jar/com/google/common/collect/CollectSpliterators.java
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
jar/com/google/common/collect/ImmutableBiMapFauxverideShim.java
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
jar/com/google/common/graph/package-info.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2015 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
jar/com/google/common/io/ReaderInputStream.java
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
jar/com/google/common/util/concurrent/InterruptibleTask.java
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
```

jar/com/google/common/hash/FarmHashFingerprint64.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/AggregateFutureState.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/AsyncCallable.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/hash/MacHashFunction.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/CombinedFuture.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/hash/LittleEndianByteArray.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/Platform.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/ConsumingQueueIterator.java  
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2021 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/ParametricNullness.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/ParametricNullness.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/html/ElementTypesAreNonnullByDefault.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/xml/ElementTypesAreNonnullByDefault.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/ElementTypesAreNonnullByDefault.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/net/ParametricNullness.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/hash/ElementTypesAreNonnullByDefault.java

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/escape/ElementTypesAreNonnullByDefault.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/eventbus/ParametricNullness.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/cache/ElementTypesAreNonnullByDefault.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/util/concurrent/ParametricNullness.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/collect/ElementTypesAreNonnullByDefault.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/escape/ParametricNullness.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/reflect/ParametricNullness.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/ElementTypesAreNonnullByDefault.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/html/ParametricNullness.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/primitives/ParametricNullness.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/reflect/ElementTypesAreNonnullByDefault.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/math/ElementTypesAreNonnullByDefault.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/cache/ParametricNullness.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/io/ParametricNullness.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/eventbus/ElementTypesAreNonnullByDefault.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/math/ParametricNullness.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/ParametricNullness.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/graph/ElementTypesAreNonnullByDefault.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/base/ElementTypesAreNonnullByDefault.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/net/ElementTypesAreNonnullByDefault.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/primitives/ElementTypesAreNonnullByDefault.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/xml/ParametricNullness.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-jar/com/google/common/hash/ParametricNullness.java

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2020 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
jar/com/google/common/util/concurrent/ServiceManagerBridge.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2009 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
jar/com/google/common/xml/XmlEscapers.java
```

```
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
jar/com/google/common/collect/DenseImmutableTable.java
```

```
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
jar/com/google/common/html/HtmlEscapers.java
```

```
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
jar/com/google/common/net/UrlEscapers.java
```

```
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
jar/com/google/common/collect/Cut.java
```

```
* /opt/cola/permits/1522658895_1672949003.9127731/0/guava-31-0-1-jre-sources-2-
```

jar/com/google/common/collect/MapMaker.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/cache/LocalCache.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/primitives/UnsignedBytes.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/Callables.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/Service.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/net/InternetDomainName.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/MapMakerInternalMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/escape/Escapers.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/cache/ReferenceEntry.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/io/ByteArrayDataOutput.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/primitives/SignedBytes.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/annotations/GwtCompatible.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/Splitter.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/escape/ArrayBasedCharEscaper.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/io/LineProcessor.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/JdkFutureAdapters.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/escape/ArrayBasedUnicodeEscaper.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/AbstractService.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/io/ByteProcessor.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/io/ByteArrayDataInput.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/AbstractIdleService.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/base/Platform.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/ForwardingListenableFuture.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/net/HostSpecifier.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-

jar/com/google/common/util/concurrent/ForwardingFluentFuture.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/escape/Platform.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/AbstractExecutionThreadService.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/ForwardingFuture.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/reflect/TypeResolver.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/escape/ArrayBasedEscaperMap.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/util/concurrent/SettableFuture.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/annotations/GwtIncompatible.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/RegularImmutableTable.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/cache/CacheBuilder.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/SparseImmutableTable.java  
No license file was found, but licenses were detected in source scan.

/\*  
\* Copyright (C) 2013 The Guava Authors  
\*  
\* Licensed under the Apache License, Version 2.0 (the "License");  
\* you may not use this file except in compliance with the License.  
\* You may obtain a copy of the License at  
\*  
\* <http://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS,  
\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
\* See the License for the specific language governing permissions and  
\* limitations under the License.  
\*/

Found in path(s):

\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/ImmutableMapEntry.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/io/MoreFiles.java  
\* /opt/cola/permits/1522658895\_1672949003.9127731/0/guava-31-0-1-jre-sources-2-  
jar/com/google/common/collect/MultimapBuilder.java

# 1.84 junit-jupiter-junit-jupiter-params 5.8.2

## 1.84.1 Available under license :

Apache License

=====

\_Version 2.0, January 2004\_

\_&lt;<<https://www.apache.org/licenses/>>>&gt;\_

### Terms and Conditions for use, reproduction, and distribution

#### 1. Definitions

License shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

Licensor shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

Legal Entity shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, control means **(i)** the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or **(ii)** ownership of fifty percent (50%) or more of the outstanding shares, or **(iii)** beneficial ownership of such entity.

You (or Your) shall mean an individual or Legal Entity exercising permissions granted by this License.

Source form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

Object form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

Work shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

Derivative Works shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.



Contribution shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, submitted means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as Not a Contribution.

Contributor shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

#### #### 2. Grant of Copyright License

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

#### #### 3. Grant of Patent License

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

#### #### 4. Redistribution

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

\* **(a)** You must give any other recipients of the Work or Derivative Works a copy of

this License; and

\* **(b)**\* You must cause any modified files to carry prominent notices stating that You changed the files; and

\* **(c)**\* You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

\* **(d)**\* If the Work includes a NOTICE text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

#### #### 5. Submission of Contributions

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

#### #### 6. Trademarks

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

#### #### 7. Disclaimer of Warranty

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied,

including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

#### #### 8. Limitation of Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

#### #### 9. Accepting Warranty or Additional Liability

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Eclipse Public License - v 2.0

=====

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

#### ### 1. Definitions

Contribution means:

\* \*\*a)\*\*\* in the case of the initial Contributor, the initial content Distributed under this Agreement, and

\* \*\*b)\*\*\* in the case of each subsequent Contributor:

\* \*\*i)\*\*\* changes to the Program, and

\* \*\*ii)\*\*\* additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of **\*\*a)\*\*** distributing or **\*\*b)\*\*** making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

### ### 2. Grant of Rights

**\*\*a)\*\*** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

**\*\*b)\*\*** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

**\*\*c)\*\*** Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

**\*\*d)\*\*** Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

**\*\*e)\*\*** Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

### ### 3. Requirements

**\*\*3.1)\*\*** If a Contributor Distributes the Program in any form, then:

**\*\*a)\*\*** the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

**\*\*b)\*\*** the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

**\*\*i)\*\*** effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

**\*\*ii)\*\*** effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

**\*\*iii)\*\*** does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

**\*\*iv)\*\*** requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

**\*\*3.2)\*\*** When the Program is Distributed as Source Code:

**\*\*a)\*\*** it must be made available under this Agreement, or if the Program **\*\*i)\*\*** is combined with other material in a separate file or files made available under a Secondary License, and **\*\*ii)\*\*** the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

**\*\*b)\*\*** a copy of this Agreement must be included with each copy of the Program.

**\*\*3.3)\*\*** Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

### ### 4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits

and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: **\*\*a)\*\*** promptly notify the Commercial Contributor in writing of such claim, and **\*\*b)\*\*** allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

#### ### 5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

#### ### 6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### ### 7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

#### #### Exhibit A - Form of Secondary Licenses Notice

> This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here }.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Open Source Licenses

=====

This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the subcomponent's license, as noted in the LICENSE-<subcomponent>.md files.

## 1.85 netty-reactive-streams-http-support 2.0.5

## 1.85.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0  
Bundle-Description: Reactive streams implementation for Netty.  
Automatic-Module-Name: com.typesafe.netty.http  
Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>  
Bundle-SymbolicName: com.typesafe.netty.reactive-streams-http  
Built-By: marcospereira  
Bnd-LastModified: 1602622977569  
Bundle-ManifestVersion: 2  
Bundle-DocURL: <http://typesafe.com/>  
Bundle-Vendor: Typesafe  
Import-Package: com.typesafe.netty;version="[2.0,3)",io.netty.buffer;version="[4.1,5)",io.netty.channel;version="[4.1,5)",io.netty.handler.codec;version="[4.1,5)",io.netty.handler.codec.http;version="[4.1,5)",io.netty.handler.codec.http.websocketx;version="[4.1,5)",io.netty.util;version="[4.1,5)",io.netty.util.concurrent;version="[4.1,5)",org.reactivestreams;version="[1.0,2)"  
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.7))"  
Tool: Bnd-3.5.0.201709291849  
Export-Package: com.typesafe.netty.http;uses:="io.netty.channel,io.netty.handler.codec.http,io.netty.handler.codec.http.websocketx,org.reactivestreams";version="2.0.5"  
Bundle-Name: Netty Reactive Streams HTTP support  
Bundle-Version: 2.0.5  
Created-By: Apache Maven Bundle Plugin  
Build-Jdk: 1.8.0\_181

Found in path(s):

\* /opt/cola/permits/1128619532\_1649176822.43/0/netty-reactive-streams-http-2-0-5-jar/META-INF/MANIFEST.MF

## 1.86 go-testify 1.7.0

### 1.86.1 Available under license :

MIT License

Copyright (c) 2012-2020 Mat Ryer, Tyler Bunnell and contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:



The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.87 pegasus-kv/thrift v0.13.0

### 1.87.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts  
as the successor of the GNU Library Public License, version 2, hence  
the version number 2.1.]

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it

does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

## GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the

Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility

is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work

under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is

normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further



restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the

ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!  
This package was debianized by Thrift Developer's <dev@thrift.apache.org>.

This package and the Debian packaging is licensed under the Apache License, see `/usr/share/common-licenses/Apache-2.0'.

The following information was copied from Apache Thrift LICENSE file.

-----  
SOFTWARE DISTRIBUTED WITH THRIFT:

The Apache Thrift software includes a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the following licenses.

-----  
Portions of the following files are licensed under the MIT License:

lib/erl/src/Makefile.am

Please see doc/otp-base-license.txt for the full terms of this license.

-----  
The following files contain some portions of code contributed under the Thrift Software License (see doc/old-thrift-license.txt), and relicensed under the Apache 2.0 License:

compiler/cpp/Makefile.am  
compiler/cpp/src/generate/t\_cpp\_generator.cc  
compiler/cpp/src/generate/t\_csharp\_generator.cc  
compiler/cpp/src/generate/t\_erl\_generator.cc  
compiler/cpp/src/generate/t\_hs\_generator.cc  
compiler/cpp/src/generate/t\_java\_generator.cc  
compiler/cpp/src/generate/t\_ocaml\_generator.cc  
compiler/cpp/src/generate/t\_perl\_generator.cc  
compiler/cpp/src/generate/t\_php\_generator.cc  
compiler/cpp/src/generate/t\_py\_generator.cc  
compiler/cpp/src/generate/t\_rb\_generator.cc  
compiler/cpp/src/generate/t\_st\_generator.cc  
compiler/cpp/src/generate/t\_xsd\_generator.cc  
compiler/cpp/src/main.cc  
compiler/cpp/src/parse/t\_field.h  
compiler/cpp/src/parse/t\_program.h  
compiler/cpp/src/platform.h  
compiler/cpp/src/thrift.l  
compiler/cpp/src/thrift.yy  
lib/csharp/src/Protocol/TBinaryProtocol.cs  
lib/csharp/src/Protocol/TField.cs  
lib/csharp/src/Protocol/TList.cs  
lib/csharp/src/Protocol/TMap.cs  
lib/csharp/src/Protocol/TMessage.cs  
lib/csharp/src/Protocol/TMessageType.cs  
lib/csharp/src/Protocol/TProtocol.cs  
lib/csharp/src/Protocol/TProtocolException.cs  
lib/csharp/src/Protocol/TProtocolFactory.cs  
lib/csharp/src/Protocol/TProtocolUtil.cs  
lib/csharp/src/Protocol/TSet.cs

lib/csharp/src/Protocol/TStruct.cs  
lib/csharp/src/Protocol/TType.cs  
lib/csharp/src/Server/TServer.cs  
lib/csharp/src/Server/TSimpleServer.cs  
lib/csharp/src/Server/TThreadPoolServer.cs  
lib/csharp/src/TApplicationException.cs  
lib/csharp/src/Thrift.csproj  
lib/csharp/src/Thrift.sln  
lib/csharp/src/TProcessor.cs  
lib/csharp/src/Transport/TServerSocket.cs  
lib/csharp/src/Transport/TServerTransport.cs  
lib/csharp/src/Transport/TSocket.cs  
lib/csharp/src/Transport/TStreamTransport.cs  
lib/csharp/src/Transport/TTransport.cs  
lib/csharp/src/Transport/TTransportException.cs  
lib/csharp/src/Transport/TTransportFactory.cs  
lib/csharp/ThriftMSBuildTask/Properties/AssemblyInfo.cs  
lib/csharp/ThriftMSBuildTask/ThriftBuild.cs  
lib/csharp/ThriftMSBuildTask/ThriftMSBuildTask.csproj  
lib/rb/lib/thrift.rb  
lib/st/README  
lib/st/thrift.st  
test/OptionalRequiredTest.cpp  
test/OptionalRequiredTest.thrift  
test/ThriftTest.thrift

-----  
For the alocal/ax\_boost\_base.m4 and contrib/fb303/aclocal/ax\_boost\_base.m4 components:

```
Copyright (c) 2007 Thomas Porschberg <thomas@randspringer.de>

Copying and distribution of this file, with or without
modification, are permitted in any medium without royalty provided
the copyright notice and this notice are preserved.
```

-----  
For the compiler/cpp/src/md5.[ch] components:

/\*

Copyright (C) 1999, 2000, 2002 Aladdin Enterprises. All rights reserved.

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

L. Peter Deutsch  
ghost@aladdin.com

\*/

-----  
For the lib/rb/setup.rb: Copyright (c) 2000-2005 Minero Aoki,  
lib/ocaml/OCamlMakefile and lib/ocaml/README-OCamlMakefile components:  
Copyright (C) 1999 - 2007 Markus Mottl

Licensed under the terms of the GNU Lesser General Public License 2.1  
(see doc/lgpl-2.1.txt for the full terms of this license)  
Tue Oct 24 12:28:44 CDT 2006

Copyright (c) <2006> <Martin J. Logan, Erlware>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software (OTP Base, fslib, G.A.S) and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT

HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE

OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the

"License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Thrift

Copyright (C) 2006 - 2019, The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<http://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but

not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their



Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.  
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

-----  
**SOFTWARE DISTRIBUTED WITH THRIFT:**

The Apache Thrift software includes a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the following licenses.

-----  
Portions of the following files are licensed under the MIT License:

lib/erl/src/Makefile.am

Please see doc/otp-base-license.txt for the full terms of this license.

-----  
For the aclocal/ax\_boost\_base.m4 and contrib/fb303/aclocal/ax\_boost\_base.m4 components:

# Copyright (c) 2007 Thomas Porschberg <thomas@randspringer.de>

#

# Copying and distribution of this file, with or without

# modification, are permitted in any medium without royalty provided  
# the copyright notice and this notice are preserved.

-----  
For the lib/nodejs/lib/thrift/json\_parse.js:

/\*

json\_parse.js

2015-05-02

Public Domain.

NO WARRANTY EXPRESSED OR IMPLIED. USE AT YOUR OWN RISK.

\*/

(By Douglas Crockford <douglas@crockford.com>)

-----  
  
Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.  
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

## APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.88 findbugs-jsr305 3.0.2

## 1.88.1 Available under license :

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

### 1. Definitions

1. "Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

2. "Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as



a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

3. "Licensor" means the individual or entity that offers the Work under the terms of this License.
4. "Original Author" means the individual or entity who created the Work.
5. "Work" means the copyrightable work of authorship offered under the terms of this License.
6. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

1. to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
2. to create and reproduce Derivative Works;
3. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;
4. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.
- 5.

For the avoidance of doubt, where the work is a musical composition:

1. Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

2. Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

6. Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following

restrictions:

1. You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(b), as requested.

2. If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

#### 5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### 7. Termination

1. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

2. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

## 8. Miscellaneous

1. Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

2. Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

3. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

4. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

5. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

# 1.89 aws-java-sdk-:::core-:::protocols-:::aws-cbor-protocol 2.17.101

## 1.89.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. **Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

#### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:  
Apache Commons Lang  
Copyright 2001-2020 The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (<https://www.apache.org/>).

# 1.90 kotlin-scripting-compiler-impl-embeddable 1.3.50

## 1.90.1 Available under license :

The version of Rhino used in GWT is licensed under a dual license, Netscape Public License 1.1 / GNU General Public License. The text of the Netscape Public License is provided below (<http://website-archive.mozilla.org/www.mozilla.org/mpl/MPL/NPL/1.1/>):

#### AMENDMENTS

The Netscape Public License Version 1.1 ("NPL") consists of the Mozilla Public License Version 1.1 with the following Amendments, including Exhibit A-Netscape Public License. Files identified with "Exhibit A-Netscape Public License" are governed by the Netscape Public License Version 1.1.



Additional Terms applicable to the Netscape Public License.

I. Effect.

These additional terms described in this Netscape Public License -- Amendments shall apply to the Mozilla Communicator client code and to all Covered Code under this License.

II. "Netscape's Branded Code" means Covered Code that Netscape distributes and/or permits others to distribute under one or more trademark(s) which are controlled by Netscape but which are not licensed for use under this License.

III. Netscape and logo.

This License does not grant any rights to use the trademarks "Netscape", the "Netscape N and horizon" logo or the "Netscape lighthouse" logo, "Netcenter", "Gecko", "Java" or "JavaScript", "Smart Browsing" even if such marks are included in the Original Code or Modifications.

IV. Inability to Comply Due to Contractual Obligation.

Prior to licensing the Original Code under this License, Netscape has licensed third party code for use in Netscape's Branded Code. To the extent that Netscape is limited contractually from making such third party code available under this License, Netscape may choose to reintegrate such code into Covered Code without being required to distribute such code in Source Code form, even if such code would otherwise be considered "Modifications" under this License.

V. Use of Modifications and Covered Code by Initial Developer.

V.1. In General.

The obligations of Section 3 apply to Netscape, except to the extent specified in this Amendment, Section V.2 and V.3.

V.2. Other Products.

Netscape may include Covered Code in products other than the Netscape's Branded Code which are released by Netscape during the two (2) years following the release date of the Original Code, without such additional products becoming subject to the terms of this License,

and may license such additional products on different terms from those contained in this License.

### V.3. Alternative Licensing.

Netscape may license the Source Code of Netscape's Branded Code, including Modifications incorporated therein, without such Netscape Branded Code becoming subject to the terms of this License, and may license such Netscape Branded Code on different terms from those contained in this License.

### VI. Litigation.

Notwithstanding the limitations of Section 11 above, the provisions regarding litigation in Section 11(a), (b) and (c) of the License shall apply to all disputes relating to this License.

### EXHIBIT A-Netscape Public License.

"The contents of this file are subject to the Netscape Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.mozilla.org/NPL/>

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is Mozilla Communicator client code, released March 31, 1998.

The Initial Developer of the Original Code is Netscape Communications Corporation. Portions created by Netscape are Copyright (C) 1998-1999 Netscape Communications Corporation. All Rights Reserved.

Contributor(s): \_\_\_\_\_.

Alternatively, the contents of this file may be used under the terms of the \_\_\_\_\_ license (the [\_\_\_\_\_] License), in which case the provisions of [\_\_\_\_\_] License are applicable instead of those above. If you wish to allow use of your version of this file only

under the terms of the [\_\_\_\_] License and not to allow others to use your version of this file under the NPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [\_\_\_\_] License. If you do not delete the provisions above, a recipient may use your version of this file under either the NPL or the [\_\_\_\_] License."

-----

## MOZILLA PUBLIC LICENSE

Version 1.1

### 1. Definitions.

1.0.1. "Commercial Use" means distribution or otherwise making the Covered Code available to a third party.

1.1. "Contributor" means each entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

1.4. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. "Executable" means Covered Code in any form other than Source Code.

1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed

herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

## 2. Source Code License.

### 2.1. The Initial Developer Grant.

The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and

(b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

(c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

## 2.2. Contributor Grant.

Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by that Contributor.

### 3. Distribution Obligations.

#### 3.1. Application of License.

The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

#### 3.2. Availability of Source Code.

Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

#### 3.3. Description of Modifications.

You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

### 3.4. Intellectual Property Matters

#### (a) Third Party Claims.

If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

#### (b) Contributor APIs.

If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

#### (c) Representations.

Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

### 3.5. Required Notices.

You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

### 3.6. Distribution of Executable Versions.

You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

### 3.7. Larger Works.



You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

#### 4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

#### 5. Application of this License.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

#### 6. Versions of the License.

##### 6.1. New Versions.

Netscape Communications Corporation ("Netscape") may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

##### 6.2. Effect of New Versions.

Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

##### 6.3. Derivative Works.

If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrases "Mozilla", "MOZILLAPL", "MOZPL",

"Netscape", "MPL", "NPL" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

## 7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

## 8. TERMINATION.

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:

(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against

such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

## 9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

## 10. U.S. GOVERNMENT END USERS.

The Covered Code is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

#### 11. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

#### 12. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

#### 13. MULTIPLE-LICENSED CODE.

Initial Developer may designate portions of the Covered Code as Multiple-Licensed. Multiple-Licensed means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the NPL or the alternative licenses, if any, specified by the Initial Developer in the file described in

Exhibit A.

EXHIBIT A -Mozilla Public License.

``The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.mozilla.org/MPL/>

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is \_\_\_\_\_.

The Initial Developer of the Original Code is \_\_\_\_\_.  
Portions created by \_\_\_\_\_ are Copyright (C) \_\_\_\_\_.  
All Rights Reserved.

Contributor(s): \_\_\_\_\_.

Alternatively, the contents of this file may be used under the terms of the \_\_\_\_\_ license (the [\_\_\_\_\_] License), in which case the provisions of [\_\_\_\_\_] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [\_\_\_\_\_] License and not to allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [\_\_\_\_\_] License. If you do not delete the provisions above, a recipient may use your version of this file under either the MPL or the [\_\_\_\_\_] License."

[NOTE: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original Code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]

=====

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies

of this license document, but changing it is not allowed.

## Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is



allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues),

conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of

this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

=====

```
/**
```

```
* Copyright 2010 Tim Down.
```

\*  
\* Licensed under the Apache License, Version 2.0 (the "License");  
\* you may not use this file except in compliance with the License.  
\* You may obtain a copy of the License at  
\*  
\* <http://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS,  
\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
\* See the License for the specific language governing permissions and  
\* limitations under the License.  
\*/

The Trove library is licensed under the Lesser GNU Public License,  
which is included with the distribution in a file called trove\_license.txt.

The PrimeFinder and HashFunctions classes in Trove are subject to the  
following license restrictions:

Copyright (c) 1999 CERN - European Organization for Nuclear Research.

Permission to use, copy, modify, distribute and sell this software and  
its documentation for any purpose is hereby granted without fee,  
provided that the above copyright notice appear in all copies and that  
both that copyright notice and this permission notice appear in  
supporting documentation. CERN makes no representations about the  
suitability of this software for any purpose. It is provided "as is"  
without expressed or implied warranty.

Copyright (c) 2005-2010 Sam Stephenson

Permission is hereby granted, free of charge, to any person obtaining a copy  
of this software and associated documentation files (the "Software"), to deal  
in the Software without restriction, including without limitation the rights  
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
copies of the Software, and to permit persons to whom the Software is  
furnished to do so, subject to the following conditions:

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE  
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE  
SOFTWARE.

/\*  
\* Copyright 2010-2018 JetBrains s.r.o.  
\*

- \* Licensed under the Apache License, Version 2.0 (the "License");
- \* you may not use this file except in compliance with the License.
- \* You may obtain a copy of the License at
- \*
- \* <http://www.apache.org/licenses/LICENSE-2.0>
- \*
- \* Unless required by applicable law or agreed to in writing, software
- \* distributed under the License is distributed on an "AS IS" BASIS,
- \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- \* See the License for the specific language governing permissions and
- \* limitations under the License.
- \*/

The Apache 2 license (given in full in LICENSE.txt) applies to all code in this repository which is copyright by JetBrains. The following sections of the repository contain third-party code, to which different licenses may apply:

### ## Kotlin Compiler

The following modules contain third-party code and are incorporated into the Kotlin compiler and/or the Kotlin IntelliJ IDEA plugin:

- Path: compiler/backend/src/org/jetbrains/kotlin/codegen/inline/MaxStackSizeAndLocalsCalculator.java
  - License: BSD ([license/third\_party/asm\_license.txt][asm])
  - Origin: Derived from ASM: a very small and fast Java bytecode manipulation framework, Copyright (c) 2000-2011 INRIA, France Telecom
- Path: compiler/backend/src/org/jetbrains/kotlin/codegen/inline/MaxLocalsCalculator.java
  - License: BSD ([license/third\_party/asm\_license.txt][asm])
  - Origin: Derived from ASM: a very small and fast Java bytecode manipulation framework, Copyright (c) 2000-2011 INRIA, France Telecom
- Path: compiler/backend/src/org/jetbrains/kotlin/codegen/optimization/common/MethodAnalyzer.kt
  - License: BSD ([license/third\_party/asm\_license.txt][asm])
  - Origin: Derived from ASM: a very small and fast Java bytecode manipulation framework, Copyright (c) 2000-2011 INRIA, France Telecom
- Path: core/reflection.jvm/src/kotlin.reflect/jvm/internal/pcollections
  - License: MIT ([license/third\_party/pcollections\_LICENSE.txt][pcollections])
  - Origin: Derived from PCollections, A Persistent Java Collections Library (<https://pcollections.org/>)
- Path: eval4j/src/org/jetbrains/eval4j/interpreterLoop.kt
  - License: BSD ([license/third\_party/asm\_license.txt][asm])
  - Origin: Derived from ASM: a very small and fast Java bytecode manipulation framework, Copyright (c) 2000-2011 INRIA, France Telecom
- Path: compiler/backend/src/org/jetbrains/kotlin/codegen/optimization/common/OptimizationBasicInterpreter.java
  - License: BSD ([license/third\_party/asm\_license.txt][asm])
  - Origin: Derived from ASM: a very small and fast Java bytecode manipulation framework, Copyright (c) 2000-

- Path: js/js.ast
  - License: BSD ([license/third\_party/dart\_LICENSE.txt][dart])
  - Origin: Originally part of the Dart compiler, (c) 2011 the Dart Project Authors,
  
- Path: js/js.inliner/src/org/jetbrains/kotlin/js/inline/FunctionInlineMutator.kt
  - License: BSD ([license/third\_party/dart\_LICENSE.txt][dart])
  - Origin: Originally part of the Dart compiler, (c) 2011 the Dart Project Authors,
  
- Path: js/js.parser/src/com/google
  - License: Netscape Public License 1.1 ([license/third\_party/rhino\_LICENSE.txt][rhino])
  - Origin: Originally part of GWT, (C) 2007-08 Google Inc., distributed under the Apache 2 license. The code is derived from Rhino, (C) 1997-1999 Netscape Communications Corporation, distributed under the Netscape Public License.
  
- Path: js/js.translator/qunit/qunit.js
  - License: MIT ([license/third\_party/qunit\_license.txt][qunit])
  - Origin: QUnit, Copyright (c) 2012 John Resig, Jrn Zaefferer,
  
- Path: libraries/stdlib/src/kotlin/collections
  - License: Apache 2 ([license/third\_party/gwt\_license.txt][gwt])
  - Origin: Derived from GWT, (C) 2007-08 Google Inc.
  
- Path: libraries/stdlib/unsigned/src/kotlin/UnsignedUtils.kt
  - License: Apache 2 ([license/third\_party/guava\_license.txt][guava])
  - Origin: Derived from Guava's UnsignedLongs, (C) 2011 The Guava Authors
  
- Path: libraries/stdlib/jvm/src/kotlin/util/MathJVM.kt
  - License: Boost Software License 1.0 ([license/third\_party/boost\_LICENSE.txt][boost])
  - Origin: Derived from boost special math functions, Copyright Eric Ford & Hubert Holin 2001.
  
- Path: libraries/stdlib/js/src/kotlin/collections
  - License: Apache 2 ([license/third\_party/gwt\_license.txt][gwt])
  - Origin: Derived from GWT, (C) 2007-08 Google Inc.
  
- Path: libraries/stdlib/js-v1/src/js/long.js
  - License: Apache 2 ([license/third\_party/closure-compiler\_LICENSE.txt][closure-compiler])
  - Origin: Google Closure Library, Copyright 2009 The Closure Library Authors
  
- Path: libraries/stdlib/js-v1/src/js/polyfills.js
  - License: Boost Software License 1.0 ([license/third\_party/boost\_LICENSE.txt][boost])
  - Origin: Derived from boost special math functions, Copyright Eric Ford & Hubert Holin 2001.
  
- Path: plugins/lint/android-annotations
  - License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
  - Origin: Copyright (C) 2011-15 The Android Open Source Project

- Path: plugins/lint/lint-api
  - License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
  - Origin: Copyright (C) 2011-15 The Android Open Source Project
- Path: plugins/lint/lint-checks
  - License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
  - Origin: Copyright (C) 2011-15 The Android Open Source Project
- Path: plugins/lint/lint-idea
  - License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
  - Origin: Copyright (C) 2011-15 The Android Open Source Project

## ## Kotlin Test Data

The following source code is used for testing the Kotlin compiler and/or plugin and is not incorporated into any distributions of the compiler, libraries or plugin:

- Path: third-party/annotations/android
  - License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
  - Origin: Copyright (C) 2011-15 The Android Open Source Project
- Path: third-party/annotations/com/android
  - License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
  - Origin: Copyright (C) 2011-15 The Android Open Source Project
- Path: third-party/annotations/org/eclipse
  - License: Eclipse Public License v1.0 ([license/third\_party/testdata/eclipse\_license.txt][eclipse])
  - Origin: Eclipse JDT, Copyright (c) 2011, 2013 Stephan Herrmann and others.
- Path: third-party/annotations/androidx
  - License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
  - Origin: Copyright (C) 2011-15 The Android Open Source Project
- Path: third-party/annotations/edu/umd/cs/findbugs
  - License: LGPL 2.1 ([license/third\_party/testdata/findbugs\_license.txt][findbugs])
  - Origin: Bytecode Analysis Framework, Copyright (C) 2005 University of Maryland
- Path: third-party/jdk8-annotations/org/eclipse
  - License: Eclipse Public License v1.0 ([license/third\_party/testdata/eclipse\_license.txt][eclipse])
  - Origin: Eclipse JDT, Copyright (c) 2011, 2013 Stephan Herrmann and others.
- Path: third-party/annotations/io/reactivex
  - License: Apache 2 ([license/third\_party/testdata/rxjava\_license.txt][rxjava])
  - Origin: RxJava, Copyright (c) 2016-present, RxJava Contributors
- Path: third-party/annotations/lombok
  - License: MIT ([license/third\_party/testdata/lombok\_license.txt][lombok])
  - Origin: Project Lombok, Copyright (C) 2009-2013 The Project Lombok Authors



- Path: idea/idea-android/tests/org/jetbrains/kotlin/android/AndroidTestBase.java
  - License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
  - Origin: Copyright (C) 2011-15 The Android Open Source Project
  
- Path: idea/testData/android/lintQuickfix/requiresApi/RequiresApi.java
  - License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
  - Origin: Copyright (C) 2011-15 The Android Open Source Project
  
- Path: idea/testData/android/lint/IntRange.java
  - License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
  - Origin: Copyright (C) 2011-15 The Android Open Source Project
  
- Path: idea/testData/android/lint/RequiresPermission.java
  - License: Apache 2 ([license/third\_party/aosp\_license.txt][aosp])
  - Origin: Copyright (C) 2011-15 The Android Open Source Project
  
- Path: libraries/tools/kotlin-gradle-plugin-integration-
 tests/src/test/resources/testProject/allOpenSpring/src/org/springframework/stereotype/Component.java
  - License: Apache 2 ([license/third\_party/testdata/spring\_license.txt][spring])
  - Origin: Spring Framework, Copyright 2002-2007 the original author or authors.
  
- Path: libraries/tools/kotlin-gradle-plugin-integration-tests/src/test/resources/testProject/AndroidDaggerProject
  - License: Apache 2 ([license/third\_party/testdata/dagger\_license.txt][dagger])
  - Origin: Dagger, Copyright (C) 2013 Square, Inc.
  
- Path: libraries/tools/kotlin-gradle-plugin-integration-tests/src/test/resources/testProject/kapt2
  - License: Apache 2 ([license/third\_party/testdata/dagger\_license.txt][dagger])
  - Origin: Dagger, Copyright (C) 2013 Square, Inc.
  
- Path: libraries/tools/kotlin-maven-plugin-test/src/it/test-allopen-
 spring/src/main/java/org/springframework/stereotype/Component.java
  - License: Apache 2 ([license/third\_party/testdata/spring\_license.txt][spring])
  - Origin: Spring Framework, Copyright 2002-2007 the original author or authors.

### ## Kotlin Tools and Libraries Tests

The following source code is used for testing the Kotlin tools and/or libraries and is not incorporated into any distributions of the tools or libraries:

- Path: libraries/tools/kotlin-gradle-plugin-integration-tests/src/test/kotlin/org/jetbrains/kotlin/gradle/PluginsDslIT.kt
  - License: Apache 2 ([license/third\_party/testdata/gradle\_license.txt][gradle])
  - Origin: Gradle, Copyright 2002-2017 Gradle, Inc.
  
- Path: libraries/tools/kotlin-gradle-plugin-integration-tests/target/test-
 classes/testProject/noArgJpa/src/javax/persistence/Entity.java
  - License: Eclipse Public License v1.0 ([license/third\_party/testdata/eclipse\_license.txt][eclipse])  
and Eclipse Distribution License - v1.0

([license/third\_party/testdata/eclipse\_distribution\_license.txt][eclipse-distribution])

- Origin: javax.persistence, Copyright (c) 2008, 2017 Sun Microsystems, Oracle Corporation.

- Path: libraries/tools/kotlin-gradle-plugin-integration-

tests/src/test/resources/testProject/noArgJpa/src/javax/persistence/Entity.java

- License: Eclipse Public License v1.0 ([license/third\_party/testdata/eclipse\_license.txt][eclipse])

and Eclipse Distribution License - v1.0

([license/third\_party/testdata/eclipse\_distribution\_license.txt][eclipse-distribution])

- Origin: javax.persistence, Copyright (c) 2008, 2017 Sun Microsystems, Oracle Corporation.

- Path: libraries/tools/kotlin-gradle-plugin-integration-

tests/src/test/resources/testProject/noArgJpa/src/javax/persistence/Embeddable.java

- License: Eclipse Public License v1.0 ([license/third\_party/testdata/eclipse\_license.txt][eclipse])

and Eclipse Distribution License - v1.0

([license/third\_party/testdata/eclipse\_distribution\_license.txt][eclipse-distribution])

- Origin: javax.persistence, Copyright (c) 2008, 2017 Sun Microsystems, Oracle Corporation.

## ## Example Code

The following code is provided as examples and is not incorporated into any distributions of the compiler, libraries or plugin:

- Path: libraries/examples/browser-example/src/js/jquery.js

- License: MIT ([license/third\_party/jquery\_license.txt][jquery])

- Origin: jQuery JavaScript Library v1.6.2, Copyright 2011, John Resig

- Path: libraries/examples/browser-example-with-library/src/js/jquery.js

- License: MIT ([license/third\_party/jquery\_license.txt][jquery])

- Origin: jQuery JavaScript Library v1.6.2, Copyright 2011, John Resig

[aosp]: third\_party/aosp\_license.txt

[asm]: third\_party/asm\_license.txt

[boost]: third\_party/boost\_LICENSE.txt

[closure-compiler]: third\_party/closure-compiler\_LICENSE.txt

[dagger]: third\_party/testdata/dagger\_license.txt

[dart]: third\_party/dart\_LICENSE.txt

[eclipse]: third\_party/testdata/eclipse\_license.txt

[eclipse-distribution]: third\_party/testdata/eclipse\_distribution\_license.txt

[findbugs]: third\_party/testdata/findbugs\_license.txt

[gradle]: third\_party/testdata/gradle\_license.txt

[guava]: third\_party/guava\_license.txt

[gwt]: third\_party/gwt\_license.txt

[jquery]: third\_party/jquery\_license.txt

[lombok]: third\_party/testdata/lombok\_license.txt

[pcollections]: third\_party/pcollections\_LICENSE.txt

[qunit]: third\_party/qunit\_license.txt

[rhino]: third\_party/rhino\_LICENSE.txt

[rxjava]: third\_party/testdata/rxjava\_license.txt

[spring]: third\_party/testdata/spring\_license.txt  
Copyright (C) 2009-2015 The Project Lombok Authors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

SCALA LICENSE

Copyright (c) 2002-2012 EPFL, Lausanne, unless otherwise specified.  
All rights reserved.

This software was developed by the Programming Methods Laboratory of the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland.

Permission to use, copy, modify, and distribute this software in source or binary form for any purpose with or without fee is hereby granted, provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the EPFL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/\* PRESENT 1 \*/

/\* PRESENT 2 \*/

// PRESENT 3

/\*\* ABSENT \*/

package/\* ABSENT 1 \*/ normal

/\* ABSENT 2 \*/

// COMMENTS: 3

ASM: a very small and fast Java bytecode manipulation framework  
Copyright (c) 2000-2005 INRIA, France Telecom  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Eclipse Distribution License - v 1.0

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
/* ABSENT */
```

```
class Some
```

```
// COMMENTS: 0
```

This license applies to all parts of Dart that are not externally maintained libraries. The external maintained libraries used by Dart are:

```
7-Zip - in third_party/7zip
JSCRE - in runtime/third_party/jscre
Ant - in third_party/apache_ant
args4j - in third_party/args4j
bzip2 - in third_party/bzip2
dromaeo - in samples/third_party/dromaeo
Eclipse - in third_party/eclipse
gsutil = in third_party/goutil
Guava - in third_party/guava
hamcrest - in third_party/hamcrest
HttpLib2 - in samples/third_party/httpLib2
JSON - in third_party/json
JUnit - in third_party/junit
Oauth - in samples/third_party/oauth2client
Rhino - in third_party/rhino
weberknecht - in third_party/weberknecht
```

The libraries may have their own licenses; we recommend you read them, as their terms may differ from the terms below.

Copyright 2012, the Dart project authors. All rights reserved.  
Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions are  
met:

- \* Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above  
copyright notice, this list of conditions and the following  
disclaimer in the documentation and/or other materials provided  
with the distribution.
- \* Neither the name of Google Inc. nor the names of its  
contributors may be used to endorse or promote products derived  
from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS  
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT  
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR  
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT  
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,  
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT  
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,  
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY  
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT  
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE  
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/\*

- \* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
  - \* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
- \*/

```
package org.jetbrains.idl2k.util
```

```
import org.xml.sax.InputSource
import java.io.File
import javax.xml.xpath.XPathFactory
```

```
fun readCopyrightNoticeFromProfile(copyrightProfile: File): String {
 val template = copyrightProfile.reader().use { reader ->
 XPathFactory.newInstance().newXPath().evaluate("/component/copyright/option[@name='notice']/@value",
 InputSource(reader))
 }
 val yearTemplate = "$today.year"
 val year = java.time.LocalDate.now().year.toString()
 assert(yearTemplate in template)

 return template.replace(yearTemplate, year).lines().joinToString("", prefix = "/*\n", postfix = " *\n") { " * $it\n" }
}
```

/\* PRESENT \*/

fun some() {}

// COMMENTS: 1

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without



modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade

names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
  
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
  
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

---

Gradle Subcomponents:

---

License for the slf4j package

---

SLF4J License

Copyright (c) 2004-2007 QOS.ch

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

These terms are identical to those of the MIT License, also called the X License or the X11 License, which is a simple, permissive non-copyleft free software license. It is deemed compatible with virtually all types of licenses, commercial or otherwise. In particular, the Free Software Foundation has declared it compatible with GNU GPL. It is also known to be approved by the Apache Software Foundation as compatible with Apache Software License.

-----  
License for the JUnit package  
-----

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

## 1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

## 2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

### 3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

- a) it complies with the terms and conditions of this Agreement; and
- b) its license agreement:
  - i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
  - ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential

damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

- a) it must be made available under this Agreement; and
- b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

#### 4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such

Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

## 5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

## 6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## 7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does

not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

-----  
License for the JCIFS package  
-----

JCIFS License

#### GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts  
as the successor of the GNU Library Public License, version 2, hence  
the version number 2.1.]

Preamble



The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a

restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and

modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

## GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a

derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2)

will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute

the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that



system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if

necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990

Ty Coon, President of Vice

That's all there is to it!

/\* PRESENT \*/

package normal

// COMMENTS: 1

Google Dart Js backend was removed - <https://code.google.com/p/dart/source/detail?r=4771>

According to <http://www.apache.org/legal/3party.html> we can include "Google Dart Js backend" in source form, because code license is "New BSD License" (Authorized License).

This part of code will be removed when kotlin will be rewritten on kotlin.

Copyright JS Foundation and other contributors, <https://js.foundation/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at <https://github.com/jquery/jquery>

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

/\* PRESENT \*/

foo()

// COMMENTS: 1

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

/\*

\* Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.

\* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

\*/

```
package org.jetbrains.kotlin.copyright
```

```
import junit.framework.AssertionFailedError
```

```
import org.jetbrains.kotlin.idea.copyright.UpdateKotlinCopyright
```

```
import org.jetbrains.kotlin.idea.test.KotlinLightCodeInsightFixtureTestCase
```

```
import org.jetbrains.kotlin.idea.test.PluginTestCaseBase
```

```
import org.jetbrains.kotlin.test.InTextDirectivesUtils
```

```
import org.junit.Assert
```

```
import java.io.File
```

```
abstract class AbstractUpdateKotlinCopyrightTest : KotlinLightCodeInsightFixtureTestCase() {
```

```
 fun doTest(path: String) {
```

```

myFixture.configureByFile(path)

val fileText = myFixture.file.text.trim()
val expectedNumberOfComments = InTextDirectivesUtils.getPrefixedInt(fileText, "// COMMENTS: ") ?: run {
 if (fileText.isNotEmpty()) {
 throw AssertionError("Every test should assert number of comments with `COMMENTS`
directive")
 } else {
 0
 }
}

val comments = UpdateKotlinCopyright.getExistentComments(myFixture.file)
for (comment in comments) {
 val commentText = comment.text
 when {
 commentText.contains("PRESENT") -> {
 }
 commentText.contains("ABSENT") -> {
 throw AssertionError("Unexpected comment found: `$commentText`")
 }
 else -> {
 throw AssertionError("A comment with bad directive found: `$commentText`")
 }
 }
}

Assert.assertEquals(
 "Wrong number of comments found:\n${comments.joinToString(separator = "\n") { it.text }}\n",
 expectedNumberOfComments, comments.size
)
}

override fun getTestDataPath() = File(PluginTestCaseBase.getTestDataPathBase(), "/copyright").path +
File.separator
}
/*
* Copyright 2000-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
*/

package tasks

import groovy.util.Node
import groovy.util.XmlParser
import org.gradle.api.DefaultTask
import org.gradle.api.Project
import org.gradle.api.tasks.Input

```

```

import org.gradle.api.tasks.InputFile
import org.gradle.api.tasks.OutputFile
import org.gradle.api.tasks.TaskAction
import java.io.File
import java.util.*

open class WriteCopyrightToFile : DefaultTask() {

 @InputFile
 var path = project.file("${project.rootDir}/.idea/copyright/apache.xml")

 @OutputFile
 var outputFile: File? = null

 @Input
 var commented: Boolean = true

 @TaskAction
 fun write() {
 if (commented) {
 outputFile!!.writeText(project.readCopyrightCommented())
 } else {
 outputFile!!.writeText(project.readCopyright())
 }
 }
}

fun Project.readCopyright(): String {
 val file = rootDir.resolve(".idea/copyright/apache.xml")

 assert(file.exists()) {
 "File $file with copyright not found"
 }

 val xmlParser = XmlParser()
 val node = xmlParser.parse(file)
 assert(node.attribute("name") == "CopyrightManager") {
 "Format changed occasionally?"
 }

 val copyrightBlock = node.children().filterIsInstance<Node>().single()
 val noticeNode = copyrightBlock.children().filterIsInstance<Node>().single { it.attribute("name") == "notice" }
 return noticeNode.attribute("value").toString().replace("$today.year",
 GregorianCalendar()[Calendar.YEAR].toString())
}

```

```
}

fun Project.readCopyrightCommented(): String {
 return "/*\n" + readCopyright().prependIndent(" * ") + "\n */"
}
}
```

GNU LESSER GENERAL PUBLIC LICENSE  
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts  
as the successor of the GNU Library Public License, version 2, hence  
the version number 2.1.]

#### Preamble

The licenses for most software are designed to take away your  
freedom to share and change it. By contrast, the GNU General Public  
Licenses are intended to guarantee your freedom to share and change  
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some  
specially designated software packages--typically libraries--of the  
Free Software Foundation and other authors who decide to use it. You  
can use it too, but we suggest you first think carefully about whether  
this license or the ordinary General Public License is the better  
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,  
not price. Our General Public Licenses are designed to make sure that  
you have the freedom to distribute copies of free software (and charge  
for this service if you wish); that you receive source code or can get  
it if you want it; that you can change the software and use pieces of  
it in new free programs; and that you are informed that you can do  
these things.

To protect your rights, we need to make restrictions that forbid  
distributors to deny you these rights or to ask you to surrender these  
rights. These restrictions translate to certain responsibilities for  
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis  
or for a fee, you must give the recipients all the rights that we gave  
you. You must make sure that they, too, receive or can get the source

code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes



a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

## GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated

```
/*
 * Copyright 2010-2017 JetBrains s.r.o.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may

distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

- e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and

all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if

written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS



## How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!  
The files in this package are taken from mozilla's Rhino project.  
See <http://www.mozilla.org/rhino/>

The files modified from Rhino 1.5R3

(<ftp://ftp.mozilla.org/pub/js/rhino15R3.zip>).

--

The contents of this package are subject to the Netscape Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.mozilla.org/NPL/>

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is Rhino code, released May 6, 1999.

The Initial Developer of the Original Code is Netscape Communications Corporation. Portions created by Netscape are Copyright (C) 1997-2000 Netscape Communications Corporation. All Rights Reserved.

Alternatively, the contents of this file may be used under the terms of the GNU Public License (the "GPL"), in which case the provisions of the GPL are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the GPL and not to allow others to use your version of this file under the NPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the GPL. If you do not delete the provisions above, a recipient may use your version of this file under either the NPL or the GPL.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

#### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2000-2018 JetBrains s.r.o.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
=====
== NOTICE file corresponding to the section 4 d of ==
== the Apache License, Version 2.0, ==
== in this case for the Kotlin Compiler distribution. ==
=====
```

Kotlin Compiler

Copyright 2010-2019 JetBrains s.r.o and respective authors and developers

Copyright (c) 2008 Harold Cooper

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Eclipse Public License, Version 1.0 (EPL-1.0)

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

## 1. DEFINITIONS

"Contribution" means:

- a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
- b) in the case of each subsequent Contributor:
  - i) changes to the Program, and
  - ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

## 2. GRANT OF RIGHTS

- a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.
- b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution.  
No hardware per se is licensed hereunder.
- c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
- d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.



### 3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

- a) it complies with the terms and conditions of this Agreement; and
- b) its license agreement:
  - i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
  - ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
  - iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
  - iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

- a) it must be made available under this Agreement; and
- b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

### 4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any

losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

- a) promptly notify the Commercial Contributor in writing of such claim,
- and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

## 5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

## 6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## 7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

The MIT License

Copyright (c) 2003, Kohsuke Kawaguchi

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright JS Foundation and other contributors, <https://js.foundation>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at <https://github.com/qunitjs/qunit>

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

## GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

#### GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of

running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library,



and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever

changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined

library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then

the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status

of all derivatives of our free software and of promoting the sharing and reuse of software generally.

## NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU  
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public  
License along with this library; if not, write to the Free Software  
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your  
school, if any, to sign a "copyright disclaimer" for the library, if  
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the  
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,  
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by  
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all  
other entities that control, are controlled by, or are under common  
control with that entity. For the purposes of this definition,  
"control" means (i) the power, direct or indirect, to cause the  
direction or management of such entity, whether by contract or  
otherwise, or (ii) ownership of fifty percent (50%) or more of the  
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity  
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.



3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed

as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.91 apache-commons-validator 1.7

## 1.91.1 Available under license :

Apache Commons Validator  
Copyright 2001-2020 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<http://www.apache.org/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## 1.92 opentest4j-opentest4j 1.2.0

### 1.92.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).



"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## 1.93 tail v1.4.8

### 1.93.1 Available under license :

# The MIT License (MIT)

# Copyright 2015 Hewlett Packard Enterprise Development LP  
Copyright (c) 2014 ActiveState

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# This source code was written by the Go contributors.

# The master list of contributors is in the main Go distribution,

# visible at <http://tip.golang.org/CONTRIBUTORS>.

Copyright (C) 2013 99designs

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2012 The Go Authors. All rights reserved.

Copyright (c) 2012-2019 fsnotify Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

tomb - support for clean goroutine termination in Go.

Copyright (c) 2010-2011 - Gustavo Niemeyer <gustavo@niemeyer.net>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# 1.94 aws-java-sdk-::-services-::-amazon-kinesis 2.17.101

## 1.94.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct



or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of

this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following

boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by Amazon Technologies, Inc (<http://www.amazon.com/>).

\*\*\*\*\*

#### THIRD PARTY COMPONENTS

\*\*\*\*\*

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from [oauth.googlecode.com](http://oauth.googlecode.com) - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang

Copyright 2001-2020 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<https://www.apache.org/>).

# 1.95 classgraph 4.8.120

## 1.95.1 Available under license :

No license file was found, but licenses were detected in source scan.

<name>The MIT License (MIT)</name>

Found in path(s):

\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/META-INF/maven/io.github.classgraph/classgraph/pom.xml

No license file was found, but licenses were detected in source scan.

/\*

\* This file is part of ClassGraph.

\*

\* Author: Michael J. Simons

\*

\* Hosted at: <https://github.com/classgraph/classgraph>

\*

\* --

\*

\* The MIT License (MIT)

\*

\* Copyright (c) 2019 Luke Hutchison

\*

\* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

\*

\* The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT

\* LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO

\* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN

\* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE

\* OR OTHER DEALINGS IN THE SOFTWARE.

\*/

Found in path(s):

\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/classloaderhandler/SpringBootRestartClassLoaderHandler.java  
No license file was found, but licenses were detected in source scan.

/\*

\* This file is part of ClassGraph.

\*

\* Author: Sergey Bespalov

\*

\* Hosted at: <https://github.com/classgraph/classgraph>

\*

\* --

\*

\* The MIT License (MIT)

\*

\* Copyright (c) 2017 Sergey Bespalov

\*

\* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

\*

\* The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT

\* LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO

\* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN

\* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE

\* OR OTHER DEALINGS IN THE SOFTWARE.

\*/

Found in path(s):

\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/classloaderhandler/WebsphereTraditionalClassLoaderHandler.java  
No license file was found, but licenses were detected in source scan.

/\*

\* This file is part of ClassGraph.

\*

\* Author: @mcollovati  
\*  
\* Hosted at: <https://github.com/classgraph/classgraph>  
\*  
\* --  
\*  
\* The MIT License (MIT)  
\*  
\* Copyright (c) 2019 @mcollovati, contributed to the ClassGraph project  
\*  
\* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated  
\* documentation files (the "Software"), to deal in the Software without restriction, including without  
\* limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of  
\* the Software, and to permit persons to whom the Software is furnished to do so, subject to the following  
\* conditions:  
\*  
\* The above copyright notice and this permission notice shall be included in all copies or substantial  
\* portions of the Software.  
\*  
\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
\* IMPLIED, INCLUDING BUT NOT  
\* LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE  
\* AND NONINFRINGEMENT. IN NO  
\* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES  
\* OR OTHER LIABILITY, WHETHER IN  
\* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION  
\* WITH THE SOFTWARE OR THE USE  
\* OR OTHER DEALINGS IN THE SOFTWARE.  
\*/

Found in path(s):

\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/classloaderhandler/QuarkusClassLoaderHandler.java

No license file was found, but licenses were detected in source scan.

/\*

\* This file is part of ClassGraph.

\*

\* Author: Luke Hutchison

\*

\* Hosted at: <https://github.com/classgraph/classgraph>

\*

\* --

\*

\* The MIT License (MIT)

\*

\* Copyright (c) 2021 Luke Hutchison

\*

\* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated  
\* documentation files (the "Software"), to deal in the Software without restriction, including without  
\* limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of  
\* the Software, and to permit persons to whom the Software is furnished to do so, subject to the following  
\* conditions:  
\*  
\* The above copyright notice and this permission notice shall be included in all copies or substantial  
\* portions of the Software.  
\*  
\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT  
\* LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE  
AND NONINFRINGEMENT. IN NO  
\* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES  
OR OTHER LIABILITY, WHETHER IN  
\* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION  
WITH THE SOFTWARE OR THE USE  
\* OR OTHER DEALINGS IN THE SOFTWARE.  
\*/

Found in path(s):

\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/classloaderhandler/CxfContainerClassLoaderHandler.java  
No license file was found, but licenses were detected in source scan.

/\*

\* This file is part of ClassGraph.

\*

\* Author: Luke Hutchison (luke.hutch@gmail.com)

\*

\* Hosted at: <https://github.com/classgraph/classgraph>

\*

\* --

\*

\* The MIT License (MIT)

\*

\* Copyright (c) 2019 Luke Hutchison

\*

\* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated  
\* documentation files (the "Software"), to deal in the Software without restriction, including without  
\* limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of  
\* the Software, and to permit persons to whom the Software is furnished to do so, subject to the following  
\* conditions:

\*

\* The above copyright notice and this permission notice shall be included in all copies or substantial  
\* portions of the Software.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT

\* LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO

\* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN

\* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE

\* OR OTHER DEALINGS IN THE SOFTWARE.

\*/

Found in path(s):

\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/ResourceList.java

No license file was found, but licenses were detected in source scan.

/\*

\* This file is part of ClassGraph.

\*

\* Author: Johno Crawford (johno@sulake.com)

\*

\* Hosted at: <https://github.com/classgraph/classgraph>

\*

\* --

\*

\* The MIT License (MIT)

\*

\* Copyright (c) 2016 Johno Crawford

\*

\* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated

\* documentation files (the "Software"), to deal in the Software without restriction, including without

\* limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of

\* the Software, and to permit persons to whom the Software is furnished to do so, subject to the following

\* conditions:

\*

\* The above copyright notice and this permission notice shall be included in all copies or substantial

\* portions of the Software.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT

\* LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO

\* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN

\* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE

\* OR OTHER DEALINGS IN THE SOFTWARE.

\*/



Found in path(s):

\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/concurrency/AutoCloseableExecutorService.java

\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/concurrency/SimpleThreadFactory.java

No license file was found, but licenses were detected in source scan.

/\*

\* This file is part of ClassGraph.

\*

\* Author: Luke Hutchison

\*

\* Hosted at: <https://github.com/classgraph/classgraph>

\*

\* --

\*

\* The MIT License (MIT)

\*

\* Copyright (c) 2019 Luke Hutchison, with significant contributions from Davy De Durlpel

\*

\* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated

\* documentation files (the "Software"), to deal in the Software without restriction, including without

\* limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of

\* the Software, and to permit persons to whom the Software is furnished to do so, subject to the following

\* conditions:

\*

\* The above copyright notice and this permission notice shall be included in all copies or substantial

\* portions of the Software.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT

\* LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO

\* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN

\* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE

\* OR OTHER DEALINGS IN THE SOFTWARE.

\*/

Found in path(s):

\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/classloaderhandler/JBossClassLoaderHandler.java

No license file was found, but licenses were detected in source scan.

/\*

\* This file is part of ClassGraph.

\*

\* Author: R. Kempees  
\*  
\* With contributions from @cpiereworld (#414)  
\*  
\* Hosted at: <https://github.com/classgraph/classgraph>  
\*  
\* --  
\*  
\* The MIT License (MIT)  
\*  
\* Copyright (c) 2017 R. Kempees (contributed to the ClassGraph project)  
\*  
\* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated  
\* documentation files (the "Software"), to deal in the Software without restriction, including without  
\* limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of  
\* the Software, and to permit persons to whom the Software is furnished to do so, subject to the following  
\* conditions:  
\*  
\* The above copyright notice and this permission notice shall be included in all copies or substantial  
\* portions of the Software.  
\*  
\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
\* IMPLIED, INCLUDING BUT NOT  
\* LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE  
\* AND NONINFRINGEMENT. IN NO  
\* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES  
\* OR OTHER LIABILITY, WHETHER IN  
\* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION  
\* WITH THE SOFTWARE OR THE USE  
\* OR OTHER DEALINGS IN THE SOFTWARE.  
\*/

Found in path(s):

\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/classloaderhandler/WebsphereLibertyClassLoaderHandler.java  
No license file was found, but licenses were detected in source scan.

/\*

\* This file is part of ClassGraph.

\*

\* Author: Harith Elrifaie

\*

\* Hosted at: <https://github.com/classgraph/classgraph>

\*

\* --

\*

\* The MIT License (MIT)

\*

\* Copyright (c) 2017 Harith Elrufaie  
\*  
\* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated  
\* documentation files (the "Software"), to deal in the Software without restriction, including without  
\* limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of  
\* the Software, and to permit persons to whom the Software is furnished to do so, subject to the following  
\* conditions:  
\*  
\* The above copyright notice and this permission notice shall be included in all copies or substantial  
\* portions of the Software.  
\*  
\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT  
\* LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE  
AND NONINFRINGEMENT. IN NO  
\* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES  
OR OTHER LIABILITY, WHETHER IN  
\* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION  
WITH THE SOFTWARE OR THE USE  
\* OR OTHER DEALINGS IN THE SOFTWARE.  
\*/

Found in path(s):

\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/classloaderhandler/FelixClassLoaderHandler.java  
No license file was found, but licenses were detected in source scan.

/\*

\* This file is part of ClassGraph.

\*

\* Author: Luke Hutchison

\*

\* Hosted at: <https://github.com/classgraph/classgraph>

\*

\* --

\*

\* The MIT License (MIT)

\*

\* Copyright (c) 2019 Luke Hutchison

\*

\* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated  
\* documentation files (the "Software"), to deal in the Software without restriction, including without  
\* limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of  
\* the Software, and to permit persons to whom the Software is furnished to do so, subject to the following  
\* conditions:

\*

\* The above copyright notice and this permission notice shall be included in all copies or substantial  
\* portions of the Software.

\*  
\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT  
\* LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE  
AND NONINFRINGEMENT. IN NO  
\* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES  
OR OTHER LIABILITY, WHETHER IN  
\* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION  
WITH THE SOFTWARE OR THE USE  
\* OR OTHER DEALINGS IN THE SOFTWARE.  
\*/

Found in path(s):

\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/AnnotationClassRef.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/classloaderhandler/OSGiDefaultClassLoaderHandler.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/ModulePathInfo.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/FieldInfo.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/ClassInfoList.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/classloaderhandler/EquinoxContextFinderClassLoaderHandler.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/recycler/RecycleOnClose.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/classpath/ClasspathOrder.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/json/JSONUtils.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/ClassRefTypeSignature.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/json/ClassFieldCache.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/concurrency/SingletonMap.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/PotentiallyUnmodifiableList.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/urls/URLPathEncoder.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/types/Parser.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/types/TypeUtils.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/BaseTypeSignature.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-

jar/nonapi/io/github/classgraph/classloaderhandler/FallbackClassLoaderHandler.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/json/ParameterizedTypeImpl.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/Utils/FileUtils.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/Utils/LogNode.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/io/github/classgraph/AnnotationParameterValue.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/io/github/classgraph/GraphvizDotfileGenerator.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/io/github/classgraph/TypeVariableSignature.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/io/github/classgraph/ClassTypeSignature.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/io/github/classgraph/TypeParameter.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/json/FieldTypeInfo.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/classpath/ModuleFinder.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/classloaderhandler/TomcatWebappClassLoaderBaseHandler.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/io/github/classgraph/ScanResultObject.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/classloaderhandler/ClassLoaderHandler.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/io/github/classgraph/ClasspathElementFileDir.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/io/github/classgraph/AnnotationEnumValue.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/scanspec/ScanSpec.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/Utils/StringUtils.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/io/github/classgraph/PackageInfoList.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/classpath/CallStackReader.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/io/github/classgraph/MappableInfoList.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/io/github/classgraph/ModuleInfo.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/json/JSONArray.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/io/github/classgraph/AnnotationInfo.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-

jar/io/github/classgraph/ClasspathElementZip.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/Utils/ReflectionUtils.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/Scanner.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/Utils/VersionFinder.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/classloaderhandler/ClassGraphClassLoaderHandler.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/json/JSONObject.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/Utils/FastPathResolver.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/concurrency/InterruptedExceptionChecker.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/ClassInfo.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/MethodTypeSignature.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/PackageInfo.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/classloaderhandler/JPMSCClassLoaderHandler.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/concurrency/WorkQueue.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/ScanResult.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/ClassGraphClassLoader.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/json/TypeResolutions.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/Utils/JarUtils.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/AnnotationInfoList.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/ModuleReaderProxy.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/fastzipfilereader/FastZipEntry.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/ArrayClassInfo.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/ModuleRef.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/ClassGraphException.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/HierarchicalTypeSignature.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-

jar/io/github/classgraph/ClasspathElement.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/io/github/classgraph/ArrayTypeSignature.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/io/github/classgraph/Classfile.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/recycler/Resetable.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/json/JSONReference.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/io/github/classgraph/HasName.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/classloaderhandler/WeblogicClassLoaderHandler.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/json/ReferenceEqualityKey.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/json/JSONParser.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/io/github/classgraph/Resource.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/io/github/classgraph/TypeArgument.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/classloaderhandler/ParentLastDelegationOrderTestClassLoaderHandler.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/classpath/ClassLoaderFinder.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/io/github/classgraph/ClasspathElementPathDir.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/classloaderhandler/PlexusClassWorldsClassRealmClassLoaderHandler.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/io/github/classgraph/ClassGraph.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/json/ClassFields.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/classloaderhandler/EquinoxClassLoaderHandler.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/classloaderhandler/URLClassLoaderHandler.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/io/github/classgraph/InfoList.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/io/github/classgraph/ReferenceTypeSignature.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/types/ParseException.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/recycler/Recycler.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-  
jar/nonapi/io/github/classgraph/fastzipfilereader/LogicalZipFile.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-

jar/io/github/classgraph/ClassRefOrTypeVariableSignature.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/classloaderhandler/ClassLoaderHandlerRegistry.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/json/JSONDeserializer.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/MethodInfoList.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/json/Id.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/MethodInfo.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/TypeSignature.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/ObjectTypedValueWrapper.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/json/JSONSerializer.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/classloaderhandler/UnoOneJarClassLoaderHandler.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/AnnotationParameterValueList.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/ModuleInfoList.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/classpath/SystemJarFinder.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/FieldInfoList.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/fastzipfilereader/PhysicalZipFile.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/Utils/CollectionUtils.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/classpath/ClassLoaderOrder.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/MethodParameterInfo.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/scanspec/AcceptReject.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/io/github/classgraph/ClasspathElementModule.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/classpath/ClasspathFinder.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/fastzipfilereader/ZipFileSlice.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/fastzipfilereader/NestedJarHandler.java  
No license file was found, but licenses were detected in source scan.

/\*



\* This file is part of ClassGraph.  
\*  
\* Author: Luke Hutchison  
\*  
\* Hosted at: <https://github.com/classgraph/classgraph>  
\*  
\* --  
\*  
\* The MIT License (MIT)  
\*  
\* Copyright (c) 2020 Luke Hutchison  
\*  
\* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated  
\* documentation files (the "Software"), to deal in the Software without restriction, including without  
\* limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of  
\* the Software, and to permit persons to whom the Software is furnished to do so, subject to the following  
\* conditions:  
\*  
\* The above copyright notice and this permission notice shall be included in all copies or substantial  
\* portions of the Software.  
\*  
\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
\* IMPLIED, INCLUDING BUT NOT  
\* LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE  
\* AND NONINFRINGEMENT. IN NO  
\* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES  
\* OR OTHER LIABILITY, WHETHER IN  
\* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION  
\* WITH THE SOFTWARE OR THE USE  
\* OR OTHER DEALINGS IN THE SOFTWARE.  
\*/

Found in path(s):

\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/fileslice/Slice.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/fileslice/ArraySlice.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/fileslice/reader/SequentialReader.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/fileslice/reader/RandomAccessArrayReader.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/fileslice/reader/RandomAccessByteBufferReader.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/fileslice/reader/ClassfileReader.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-jar/nonapi/io/github/classgraph/fileslice/reader/RandomAccessFileChannelReader.java  
\* /opt/cola/permits/1411867118\_1662683845.1356816/0/classgraph-4-8-120-sources-

```
jar/nonapi/io/github/classgraph/fileslice/reader/RandomAccessReader.java
* /opt/cola/permits/1411867118_1662683845.1356816/0/classgraph-4-8-120-sources-
jar/nonapi/io/github/classgraph/fileslice/FileSlice.java
* /opt/cola/permits/1411867118_1662683845.1356816/0/classgraph-4-8-120-sources-
jar/nonapi/io/github/classgraph/fileslice/PathSlice.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* This file is part of ClassGraph.
*
* Author: @jacobg on GitHub
*
* Hosted at: https://github.com/classgraph/classgraph
*
* --
*
* The MIT License (MIT)
*
* Copyright (c) 2019 @jacobg, Luke Hutchison
*
* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
* documentation files (the "Software"), to deal in the Software without restriction, including without
* limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
* the Software, and to permit persons to whom the Software is furnished to do so, subject to the following
* conditions:
*
* The above copyright notice and this permission notice shall be included in all copies or substantial
* portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT
* LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
AND NONINFRINGEMENT. IN NO
* EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES
OR OTHER LIABILITY, WHETHER IN
* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE
* OR OTHER DEALINGS IN THE SOFTWARE.
*/
```

Found in path(s):

```
* /opt/cola/permits/1411867118_1662683845.1356816/0/classgraph-4-8-120-sources-
jar/nonapi/io/github/classgraph/classloaderhandler/AntClassLoaderHandler.java
```

## 1.96 pgv-java-stubs 0.6.1

## 1.96.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
  
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
  
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

## 1.97 go-humanize v1.0.0

### 1.97.1 Available under license :

Copyright (c) 2005-2008 Dustin Sallings <dustin@spy.net>

Permission is hereby granted, free of charge, to any person obtaining a copy  
of this software and associated documentation files (the "Software"), to deal  
in the Software without restriction, including without limitation the rights  
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
copies of the Software, and to permit persons to whom the Software is  
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in  
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE  
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE  
SOFTWARE.

<<http://www.opensource.org/licenses/mit-license.php>>

## 1.98 protobuf v1.27.1

### 1.98.1 Available under license :

Copyright (c) 2018 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions are  
met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

- # This source code was written by the Go contributors.
- # The master list of contributors is in the main Go distribution,
- # visible at <https://tip.golang.org/CONTRIBUTORS>.

## 1.99 commons-logging 1.2

### 1.99.1 Available under license :

- ```
/*
 * Licensed to the Apache Software Foundation (ASF) under one or more
 * contributor license agreements. See the NOTICE file distributed with
 * this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0
 * (the "License"); you may not use this file except in compliance with
 * the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Apache License
Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Commons Logging

Copyright 2003-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

1.100 x-xerrors 20200804-snapshot-5ec99f83

1.100.1 Available under license :

Copyright (c) 2019 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.101 aws-secretsmanager-caching-java 1.0.1

1.101.1 Available under license :

AWS Secrets Manager Java caching client

Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the

same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.102 perfmark-perfmark-api 0.23.0

1.102.1 Available under license :

Copyright 2019 Google LLC

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This product contains a modified portion of 'Catapult', an open source Trace Event viewer for Chrome, Linux, and Android applications, which can be obtained at:

* LICENSE:

* [traceviewer/src/main/resources/io/perfmark/traceviewer/third_party/catapult/LICENSE](#) (New BSD License)

* HOMEPAGE:

* <https://github.com/catapult-project/catapult>

This product contains a modified portion of 'Polymer', a library for Web

Components, which can be obtained at:

* LICENSE:

* [traceviewer/src/main/resources/io/perfmark/traceviewer/third_party/polymer/LICENSE](https://github.com/Polymer/polymer/blob/master/src/main/resources/io/perfmark/traceviewer/third_party/polymer/LICENSE) (New BSD License)

* HOMEPAGE:

* <https://github.com/Polymer/polymer>

Copyright 2015 The Chromium Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of catapult nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or,

within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all

other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
// Copyright (c) 2012 The Polymer Authors. All rights reserved.  
//  
// Redistribution and use in source and binary forms, with or without  
// modification, are permitted provided that the following conditions are  
// met:  
//  
// * Redistributions of source code must retain the above copyright  
// notice, this list of conditions and the following disclaimer.
```

```
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
/*
 * Copyright 2019 Google LLC
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

1.103 j2objc-annotations 1.3

1.103.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *

```


- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.

*/

Found in path(s):

- * /opt/cola/permits/1131003150_1612875443.99/0/j2objc-annotations-1-3-sources-3-jar/com/google/j2objc/annotations/RetainedWith.java
- * /opt/cola/permits/1131003150_1612875443.99/0/j2objc-annotations-1-3-sources-3-jar/com/google/j2objc/annotations/Property.java
- * /opt/cola/permits/1131003150_1612875443.99/0/j2objc-annotations-1-3-sources-3-jar/com/google/j2objc/annotations/LoopTranslation.java
- * /opt/cola/permits/1131003150_1612875443.99/0/j2objc-annotations-1-3-sources-3-jar/com/google/j2objc/annotations/ObjectiveCName.java
- * /opt/cola/permits/1131003150_1612875443.99/0/j2objc-annotations-1-3-sources-3-jar/com/google/j2objc/annotations/ReflectionSupport.java
- * /opt/cola/permits/1131003150_1612875443.99/0/j2objc-annotations-1-3-sources-3-jar/com/google/j2objc/annotations/RetainedLocalRef.java
- * /opt/cola/permits/1131003150_1612875443.99/0/j2objc-annotations-1-3-sources-3-jar/com/google/j2objc/annotations/J2ObjCIncompatible.java

No license file was found, but licenses were detected in source scan.

/*

- * Copyright 2012 Google Inc. All Rights Reserved.
- *
- * Licensed under the Apache License, Version 2.0 (the "License");
- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.

*/

Found in path(s):

- * /opt/cola/permits/1131003150_1612875443.99/0/j2objc-annotations-1-3-sources-3-jar/com/google/j2objc/annotations/Weak.java
- * /opt/cola/permits/1131003150_1612875443.99/0/j2objc-annotations-1-3-sources-3-jar/com/google/j2objc/annotations/AutoreleasePool.java
- * /opt/cola/permits/1131003150_1612875443.99/0/j2objc-annotations-1-3-sources-3-jar/com/google/j2objc/annotations/WeakOuter.java

1.104 wire-protocol-buffer-runtime 3.7.1

1.104.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2015 Square Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-
jar/commonMain/com/squareup/wire/internal/MutableOnWriteList.kt
* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-
jar/jvmMain/com/squareup/wire/MessageSerializedForm.kt
* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-
jar/commonMain/com/squareup/wire/ProtoAdapter.kt
* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-
jar/jvmMain/com/squareup/wire/internal/FieldBinding.kt
* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-
jar/jvmMain/com/squareup/wire/ProtoAdapter.kt
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2013 Square Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/commonMain/com/squareup/wire/WireField.kt
* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/commonMain/com/squareup/wire/internal/RuntimeMessageAdapter.kt
* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/jvmMain/com/squareup/wire/Message.kt
* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/jvmMain/com/squareup/wire/Wire.kt
* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/jvmMain/com/squareup/wire/RuntimeEnumAdapter.kt
* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/commonMain/com/squareup/wire/internal/ImmutableList.kt
* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/jvmMain/com/squareup/wire/internal/InternalJvm.kt
* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/commonMain/com/squareup/wire/FieldEncoding.kt
* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/commonMain/com/squareup/wire/WireEnum.kt
* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/commonMain/com/squareup/wire/Message.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2013 Square Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

// Copyright 2013 Google Inc. All rights reserved.

// Redistribution and use in source and binary forms, with or without

// modification, are permitted provided that the following conditions are

// * Redistributions of source code must retain the above copyright

// notice, this list of conditions and the following disclaimer.

// * Redistributions in binary form must reproduce the above

// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// * Neither the name of Google Inc. nor the names of its
// this software without specific prior written permission.

Found in path(s):

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-
jar/commonMain/com/squareup/wire/ProtoWriter.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2016 Square Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-
jar/jvmMain/com/squareup/wire/EnumAdapter.kt

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-
jar/jvmMain/com/squareup/wire/AndroidMessage.kt

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-
jar/commonMain/com/squareup/wire/EnumAdapter.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2021 Square Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/jvmMain/com/squareup/wire/internal/reflection.kt

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/commonMain/com/squareup/wire/internal/MessageBinding.kt

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/jvmMain/com/squareup/wire/internal/OneOfBinding.kt

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/commonMain/com/squareup/wire/internal/FieldOrOneOfBinding.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2020 Square Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/jvmMain/com/squareup/wire/internal/JsonIntegration.kt

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/commonMain/com/squareup/wire/Instant.kt

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/jvmMain/com/squareup/wire/Duration.kt

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/commonMain/com/squareup/wire/AnyMessage.kt

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/commonMain/com/squareup/wire/OneOf.kt

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/jvmMain/com/squareup/wire/Instant.kt

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/jvmMain/com/squareup/wire/internal/InstantJsonFormatter.kt

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/jvmMain/com/squareup/wire/internal/EnumJsonFormatter.kt

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-

jar/commonMain/com/squareup/wire/Duration.kt
* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-
jar/commonMain/com/squareup/wire/Syntax.kt
* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-
jar/commonMain/com/squareup/wire/WireEnumConstant.kt
* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-
jar/jvmMain/com/squareup/wire/internal/JsonFormatter.kt
* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-
jar/jvmMain/com/squareup/wire/internal/DurationJsonFormatter.kt
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2019 Square, Inc.  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

Found in path(s):
* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-
jar/commonMain/com/squareup/wire/internal/Util.kt
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (c) 2016, the R8 project authors.  
* All rights reserved.  
*  
* Redistribution and use in source and binary forms, with or without  
* modification, are permitted provided that the following conditions are met:  
*  
* * Redistributions of source code must retain the above copyright notice, this  
* list of conditions and the following disclaimer.  
*  
* * Redistributions in binary form must reproduce the above copyright notice,  
* this list of conditions and the following disclaimer in the documentation  
* and/or other materials provided with the distribution.  
*  
* * Neither the name of Google Inc. nor the names of its  
* contributors may be used to endorse or promote products derived from
```

* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-
jar/commonMain/com/squareup/wire/internal/MathMethods.kt

No license file was found, but licenses were detected in source scan.

```
// Copyright 2013 Google Inc. All rights reserved.  
// Redistribution and use in source and binary forms, with or without  
// modification, are permitted provided that the following conditions are  
// * Redistributions of source code must retain the above copyright  
// notice, this list of conditions and the following disclaimer.  
// * Redistributions in binary form must reproduce the above  
// copyright notice, this list of conditions and the following disclaimer  
// in the documentation and/or other materials provided with the  
// * Neither the name of Google Inc. nor the names of its  
// this software without specific prior written permission.
```

Found in path(s):

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-
jar/commonMain/com/squareup/wire/ProtoReader.kt

No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright 2019 Square Inc.  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and
```

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/commonMain/com/squareup/wire/internal/-Platform.kt

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/commonMain/com/squareup/wire/Service.kt

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/jvmMain/com/squareup/wire/MessageSource.kt

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/jvmMain/com/squareup/wire/internal/-Platform.kt

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/commonMain/com/squareup/wire/MessageSink.kt

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/commonMain/com/squareup/wire/MessageSource.kt

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/jvmMain/com/squareup/wire/MessageSink.kt

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/commonMain/com/squareup/wire/WireRpc.kt

* /opt/cola/permits/1526005717_1673464893.6445808/0/wire-runtime-3-7-1-sources-jar/commonMain/com/squareup/wire/internal/Internal.kt

1.105 apache-log4j-slf4j-binding 2.19.0

1.105.1 Available under license :

Apache Log4j SLF4J Binding

Copyright 1999-2022 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (<http://www.apache.org/>).

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a

result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.106 golang-mock v1.6.0

1.106.1 Available under license :

```
# This is the official list of people who can contribute (and typically
# have contributed) code to the gomock repository.
# The AUTHORS file lists the copyright holders; this file
# lists people. For example, Google employees are listed here
# but not in AUTHORS, because Google holds the copyright.
#
# The submission process automatically checks to make sure
# that people submitting code are listed in this file (by email address).
#
# Names should be added to this file only after verifying that
# the individual or the individual's organization has agreed to
# the appropriate Contributor License Agreement, found here:
#
#   http://code.google.com/legal/individual-cla-v1.0.html
#   http://code.google.com/legal/corporate-cla-v1.0.html
#
# The agreement for individuals can be filled out on the web.
#
# When adding J Random Contributor's name to this file,
# either J's name or J's organization's name should be
# added to the AUTHORS file, depending on whether the
# individual or corporate CLA was used.

# Names should be added to this file like so:
#   Name <email address>
#
# An entry with two email addresses specifies that the
# first address should be used in the submit logs and
# that the second address should be recognized as the
# same person when interacting with Rietveld.

# Please keep the list sorted.

Aaron Jacobs <jacobsa@google.com> <aaronjjacobs@gmail.com>
Alex Reece <awreece@gmail.com>
David Symonds <dsymonds@golang.org>
Ryan Barrett <ryanb@google.com>
package empty_interface

//go:generate mockgen -package empty_interface -destination mock.go -source input.go -
copyright_file=mock_copyright_header

type Empty interface{ }
// This is a mock copyright header.
//
// Lorem ipsum dolor sit amet, consectetur adipiscing elit,
// sed do eiusmod tempor incididunt ut labore et dolore magna
```

```

// aliqua. Velit ut tortor pretium viverra suspendisse potenti.
//

// Code generated by MockGen. DO NOT EDIT.
// Source: input.go

// Package empty_interface is a generated GoMock package.
package empty_interface

import (
    gomock "github.com/golang/mock/gomock"
)

// MockEmpty is a mock of Empty interface.
type MockEmpty struct {
    ctrl    *gomock.Controller
    recorder *MockEmptyMockRecorder
}

// MockEmptyMockRecorder is the mock recorder for MockEmpty.
type MockEmptyMockRecorder struct {
    mock *MockEmpty
}

// NewMockEmpty creates a new mock instance.
func NewMockEmpty(ctrl *gomock.Controller) *MockEmpty {
    mock := &MockEmpty{ctrl: ctrl}
    mock.recorder = &MockEmptyMockRecorder{mock}
    return mock
}

// EXPECT returns an object that allows the caller to indicate expected use.
func (m *MockEmpty) EXPECT() *MockEmptyMockRecorder {
    return m.recorder
}

```

This is a mock copyright header.

Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua. Velit ut tortor pretium viverra suspendisse potenti.

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of

the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

1.107 io-grpc-grpc-context 1.39.0

1.107.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object

form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a

file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2016-2020 Istio Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Envoy

Copyright 2016-2019 Envoy Project Authors

Licensed under Apache License 2.0. See LICENSE for terms.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner].

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

protoc-gen-validate

Copyright 2019 Envoy Project Authors

Licensed under Apache License 2.0. See LICENSE for terms.

zero-allocation-hashing

Copyright 2015 Higher Frequency Trading <http://www.higherfrequencytrading.com>

Licensed under Apache License 2.0. See LICENSE for terms.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the

Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

/*
* Copyright 2015 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Copyright 2014 The gRPC Authors

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This product contains a modified portion of 'OkHttp', an open source
HTTP & SPDY client for Android and Java applications, which can be obtained
at:

- * LICENSE:
 - * [okhttp/third_party/okhttp/LICENSE](https://github.com/square/okhttp) (Apache License 2.0)
- * HOMEPAGE:
 - * <https://github.com/square/okhttp>
- * LOCATION_IN_GRPC:
 - * [okhttp/third_party/okhttp](https://github.com/square/okhttp)

This product contains a modified portion of 'Envoy', an open source
cloud-native high-performance edge/middle/service proxy, which can be
obtained at:

- * LICENSE:
 - * [xds/third_party/envoy/LICENSE](https://github.com/envoyproxy/envoy) (Apache License 2.0)

- * NOTICE:
 - * xds/third_party/envoy/NOTICE
- * HOMEPAGE:
 - * <https://www.envoyproxy.io>
- * LOCATION_IN_GRPC:
 - * xds/third_party/envoy

This product contains a modified portion of 'protoc-gen-validate (PGV)', an open source protoc plugin to generate polyglot message validators, which can be obtained at:

- * LICENSE:
 - * xds/third_party/protoc-gen-validate/LICENSE (Apache License 2.0)
- * NOTICE:
 - * xds/third_party/protoc-gen-validate/NOTICE
- * HOMEPAGE:
 - * <https://github.com/envoyproxy/protoc-gen-validate>
- * LOCATION_IN_GRPC:
 - * xds/third_party/protoc-gen-validate

This product contains a modified portion of 'udpa', an open source universal data plane API, which can be obtained at:

- * LICENSE:
 - * xds/third_party/udpa/LICENSE (Apache License 2.0)
- * HOMEPAGE:
 - * <https://github.com/cncf/udpa>
- * LOCATION_IN_GRPC:
 - * xds/third_party/udpa

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the

direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.108 aws-glue-schema-registry-build-tools

1.1.9

1.108.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<!--  
/*  
* Copyright 2020 Amazon.com, Inc. or its affiliates.  
* Licensed under the Apache License, Version 2.0 (the  
* "License"); you may not use this file except in compliance  
* with the License. You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/  
-->
```

Found in path(s):

```
* /opt/cola/permits/1526005837_1673043450.4296443/0/schema-registry-build-tools-1-1-9-jar/META-INF/maven/software.amazon.glue/schema-registry-build-tools/pom.xml
```

No license file was found, but licenses were detected in source scan.

```
<!--  
/*  
* Copyright 2019 Amazon.com, Inc. or its affiliates.  
* Licensed under the Apache License, Version 2.0 (the  
* "License"); you may not use this file except in compliance  
* with the License. You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/  
-->
```

Found in path(s):

```
* /opt/cola/permits/1526005837_1673043450.4296443/0/schema-registry-build-tools-1-1-9-jar/suppressions.xml  
* /opt/cola/permits/1526005837_1673043450.4296443/0/schema-registry-build-tools-1-1-9-jar/checkstyle.xml
```

1.109 netty-project 4.1.68.Final

1.109.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<!--
~ Copyright 2014 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ https://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.
-->
```

Found in path(s):

```
* /opt/cola/permits/1331306087_1662690272.2162187/0/netty-transport-native-epoll-4-1-68-final-jar/META-INF/maven/io.netty/netty-transport-native-epoll/pom.xml
```

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Implementation-Title: Netty/Transport/Native/Epoll

Bundle-Description: Netty is an asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers and clients.

Automatic-Module-Name: io.netty.transport.epoll

Bundle-License: <https://www.apache.org/licenses/LICENSE-2.0>

Bundle-SymbolicName: io.netty.transport-native-epoll

Implementation-Version: 4.1.68.Final

Built-By: root

Bnd-LastModified: 1631193593222

Bundle-ManifestVersion: 2

Implementation-Vendor-Id: io.netty

Bundle-DocURL: <https://netty.io/>

Bundle-Vendor: The Netty Project

Import-Package: io.netty.buffer;version="[4.1,5)",io.netty.channel,io.netty.channel.socket;version="[4.1,5)",io.netty.channel.unix;version="[4.1,5)",io.netty.util;version="[4.1,5)",io.netty.util.collection;version="[4.1,5)",io.netty.util.concurrent;version="[4.1,5)",io.netty.util.internal;version="[4.1,5)",io.netty.util.internal.logging;version="[4.1,5)",sun.nio.ch;resolution:=optional,org.eclipse.jetty.npn;vers

ion="[1,2]";resolution:=optional,org.eclipse.jetty.alpn;version="[1,2
)];resolution:=optional
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.6))"
Tool: Bnd-2.4.1.201501161923
Implementation-Vendor: The Netty Project
Export-Package: io.netty.channel.epoll;uses:="io.netty.buffer,io.netty
.channel,io.netty.channel.socket,io.netty.channel.unix,io.netty.util,
io.netty.util.concurrent";version="4.1.68"
Bundle-Name: Netty/Transport/Native/Epoll
Bundle-Version: 4.1.68.Final
Created-By: Apache Maven Bundle Plugin
Build-Jdk: 1.8.0_292
Implementation-URL: https://netty.io/netty-transport-native-epoll/

Found in path(s):

* /opt/cola/permits/1331306087_1662690272.2162187/0/netty-transport-native-epoll-4-1-68-final-jar/META-INF/MANIFEST.MF

1.110 gomemcache 20190913-snapshot-a41fca85

1.110.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.111 fsnotify-fsnotify v1.4.9

1.111.1 Available under license :

Copyright (c) 2012 The Go Authors. All rights reserved.

Copyright (c) 2012-2019 fsnotify Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.112 software-amazon-ion-java 1.0.2

1.112.1 Available under license :

Amazon Ion Java

Copyright 2007-2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.113 pgv-java-grpc-interceptors 0.6.1

1.113.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.114 testcontainers-::-localstack 1.17.3

1.114.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.115 caffeine-cache 3.0.3

1.115.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/MpscGrowableArrayQueue.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2014 Ben Manes. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/Ticker.java
```

```
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/stats/CacheStats.java
```

```
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/LinkedDeque.java
```

* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/Policy.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/RemovalListener.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/stats/DisabledStatsCounter.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/Weigher.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/AsyncLoadingCache.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/BoundedLocalCache.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/Cache.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/RemovalCause.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/WriteOrderDeque.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/UnboundedLocalCache.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/AccessOrderDeque.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/stats/StatsCounter.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/stats/ConcurrentStatsCounter.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/Caffeine.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/LoadingCache.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/AbstractLinkedDeque.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/CacheLoader.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2015 Ben Manes. All Rights Reserved.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and
* limitations under the License.
*/
/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* <http://creativecommons.org/publicdomain/zero/1.0/>
*/

Found in path(s):

* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/StripedBuffer.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2017 Ben Manes. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/Expiry.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/TimerWheel.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2018 Ben Manes. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/AsyncCache.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/LocalAsyncCache.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2016 Ben Manes. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/AsyncCacheLoader.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/CaffeineSpec.java
No license file was found, but licenses were detected in source scan.

// Copyright 2021 Ben Manes. All Rights Reserved.
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
// <http://www.apache.org/licenses/LICENSE-2.0>
// distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):

* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/SSLMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/SSLSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/WILMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWAWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSARMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWAWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSAWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/PDMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSARMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDARMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDAWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDARMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/SISMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PD.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/FDAWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSARMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDAMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSAWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/SSMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDAWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSAWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDAMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSAWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDAWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/SILMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDAWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/SSSMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSAWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLSMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWARMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDAWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWAMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSL.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/SSLSMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSAWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/SISMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDARMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSARMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLSMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/PDAWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSL.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLSMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWAWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSAWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLSMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/SIL.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWAWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDAMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDARMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/SSSMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWAWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/LocalCacheFactory.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSAMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/WSLMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLSMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLSMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/WILSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/FD.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SI.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WI.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWARMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/WILSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWAMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/FDR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWAMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSAMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSAMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/WILWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWARMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWAMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSAWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIL.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/PDWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWAWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSAMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDAMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/NodeFactory.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWAWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/WILSMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWARMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/PWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDAWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSLS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSLAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/SIMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWAWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMSR.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2019 Ben Manes. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/Scheduler.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/Pacer.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2015 Ben Manes. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FrequencySketch.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/Buffer.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/Async.java
```


* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/LocalAsyncLoadingCache.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/References.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/stats/package-info.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/LocalLoadingCache.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/WriteThroughEntry.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/Node.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/BoundedBuffer.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/package-info.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/SerializationProxy.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/LocalCache.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/stats/GuardedStatsCounter.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/LocalManualCache.java

1.116 aws-java-sdk-::-core-::-protocols-::-

aws-json-protocol 2.17.122

1.116.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by
Amazon Technologies, Inc (<http://www.amazon.com/>).

THIRD PARTY COMPONENTS

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:
Apache Commons Lang
Copyright 2001-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

1.117 asm 9.1

1.117.1 Available under license :

No license file was found, but licenses were detected in source scan.

2011 INRIA, France Telecom

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. Neither the name of the copyright holders nor the names of its

* contributors may be used to endorse or promote products derived from

* this software without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/signature/package.html

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/package.html

No license file was found, but licenses were detected in source scan.

// All rights reserved.

// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:

// 1. Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.

// 2. Redistributions in binary form must reproduce the above copyright
// notice, this list of conditions and the following disclaimer in the
// documentation and/or other materials provided with the distribution.

// 3. Neither the name of the copyright holders nor the names of its
// this software without specific prior written permission.

Found in path(s):

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/TypePath.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Frame.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/RecordComponentWriter.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/AnnotationVisitor.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/ModuleVisitor.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/ModuleWriter.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/CurrentFrame.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/AnnotationWriter.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/ConstantDynamic.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/ClassWriter.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/signature/SignatureVisitor.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Attribute.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Opcodes.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Handle.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/MethodTooLargeException.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-

jar/org/objectweb/asm/signature/SignatureReader.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/TypeReference.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Context.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Label.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Type.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-
jar/org/objectweb/asm/signature/SignatureWriter.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/FieldVisitor.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Handler.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/SymbolTable.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/ClassReader.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Edge.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-
jar/org/objectweb/asm/RecordComponentVisitor.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Constants.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Symbol.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/FieldWriter.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/MethodWriter.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/MethodVisitor.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-
jar/org/objectweb/asm/ClassTooLargeException.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/ByteVector.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/ClassVisitor.java

1.118 aws-event-stream 1.0.1

1.118.1 Available under license :

MIT License

Copyright (c) 2018 Lifion, by ADP

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

SOFTWARE.

1.119 apache-log4j-api 2.17.1

1.119.1 Available under license :

Apache Log4j API

Copyright 1999-1969 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.120 animal-sniffer-annotation 1.19

1.120.1 Available under license :

No license file was found, but licenses were detected in source scan.

The MIT License

Copyright (c) 2009 codehaus.org.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Found in path(s):

* /opt/cola/permits/1127519953_1611825926.02/0/animal-sniffer-annotations-1-19-sources-jar/META-INF/maven/org.codehaus.mojo/animal-sniffer-annotations/pom.xml

No license file was found, but licenses were detected in source scan.

/*

* The MIT License

*

* Copyright (c) 2008 Kohsuke Kawaguchi and codehaus.org.

*

* Permission is hereby granted, free of charge, to any person obtaining a copy

* of this software and associated documentation files (the "Software"), to deal

* in the Software without restriction, including without limitation the rights

* to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

* copies of the Software, and to permit persons to whom the Software is

* furnished to do so, subject to the following conditions:

*

* The above copyright notice and this permission notice shall be included in

* all copies or substantial portions of the Software.

*

* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

* AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

* LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

* OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN

* THE SOFTWARE.

*

*/

Found in path(s):

* /opt/cola/permits/1127519953_1611825926.02/0/animal-sniffer-annotations-1-19-sources-jar/org/codehaus/mojo/animal_sniffer/IgnoreJRERequirement.java

1.121 beorn7-perks v1.0.1

1.121.1 Available under license :

Copyright (C) 2013 Blake Mizerany

Permission is hereby granted, free of charge, to any person obtaining

a copy of this software and associated documentation files (the

"Software"), to deal in the Software without restriction, including

without limitation the rights to use, copy, modify, merge, publish,

distribute, sublicense, and/or sell copies of the Software, and to

permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.122 protobuf-java 3.19.4

1.122.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
// Copyright 2008 Google Inc. All rights reserved.  
// Redistribution and use in source and binary forms, with or without  
// modification, are permitted provided that the following conditions are  
// * Redistributions of source code must retain the above copyright  
// notice, this list of conditions and the following disclaimer.  
// * Redistributions in binary form must reproduce the above  
// copyright notice, this list of conditions and the following disclaimer  
// in the documentation and/or other materials provided with the  
// * Neither the name of Google Inc. nor the names of its  
// this software without specific prior written permission.
```

Found in path(s):

```
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/google/protobuf/descriptor.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/google/protobuf/duration.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/google/protobuf/struct.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/google/protobuf/field_mask.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/google/protobuf/compiler/plugin.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/google/protobuf/empty.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/google/protobuf/type.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/google/protobuf/any.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/google/protobuf/wrappers.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/google/protobuf/timestamp.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/google/protobuf/source_context.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/google/protobuf/api.proto
```

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0
Automatic-Module-Name: com.google.protobuf
Bnd-LastModified: 1643389670477
Build-Jdk: 1.8.0_181-google-v7
Built-By: acozzette
Bundle-Description: Core Protocol Buffers library. Protocol Buffers are a way of encoding structured data in an efficient yet extensible format.
Bundle-DocURL: <https://developers.google.com/protocol-buffers/>
Bundle-License: <https://opensource.org/licenses/BSD-3-Clause>
Bundle-ManifestVersion: 2
Bundle-Name: Protocol Buffers [Core]
Bundle-SymbolicName: com.google.protobuf
Bundle-Version: 3.19.4
Created-By: Apache Maven Bundle Plugin
Export-Package: com.google.protobuf;version="3.19.4"
Import-Package: sun.misc;resolution:=optional,com.google.protobuf;version="[3.19,4)"
Require-Capability: osgi.ee;filter:="(&(osgi.ee=JavaSE)(version=1.7))"
Tool: Bnd-3.0.0.201509101326

Found in path(s):

* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/META-INF/MANIFEST.MF

1.123 checker-qual 3.15.0

1.123.1 Available under license :

Checker Framework qualifiers

Copyright 2004-present by the Checker Framework developers

MIT License:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.124 netty-codec-smtp 4.1.85.Final

1.124.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<!--
~ Copyright 2016 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ https://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.
-->
```

Found in path(s):

```
* /opt/cola/permits/1498798484_1670284125.032224/0/netty-codec-smtp-4-1-85-final-sources-1-jar/META-INF/maven/io.netty/netty-codec-smtp/pom.xml
```

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2016 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
```

Found in path(s):

- * /opt/cola/permits/1498798484_1670284125.032224/0/netty-codec-smtp-4-1-85-final-sources-1-jar/io/netty/handler/codec/smtp/SntpRequest.java
- * /opt/cola/permits/1498798484_1670284125.032224/0/netty-codec-smtp-4-1-85-final-sources-1-jar/io/netty/handler/codec/smtp/DefaultSntpContent.java
- * /opt/cola/permits/1498798484_1670284125.032224/0/netty-codec-smtp-4-1-85-final-sources-1-jar/io/netty/handler/codec/smtp/SntpUtils.java
- * /opt/cola/permits/1498798484_1670284125.032224/0/netty-codec-smtp-4-1-85-final-sources-1-jar/io/netty/handler/codec/smtp/SntpContent.java
- * /opt/cola/permits/1498798484_1670284125.032224/0/netty-codec-smtp-4-1-85-final-sources-1-jar/io/netty/handler/codec/smtp/SntpResponseDecoder.java
- * /opt/cola/permits/1498798484_1670284125.032224/0/netty-codec-smtp-4-1-85-final-sources-1-jar/io/netty/handler/codec/smtp/SntpRequests.java
- * /opt/cola/permits/1498798484_1670284125.032224/0/netty-codec-smtp-4-1-85-final-sources-1-jar/io/netty/handler/codec/smtp/SntpRequestEncoder.java
- * /opt/cola/permits/1498798484_1670284125.032224/0/netty-codec-smtp-4-1-85-final-sources-1-jar/io/netty/handler/codec/smtp/DefaultSntpRequest.java
- * /opt/cola/permits/1498798484_1670284125.032224/0/netty-codec-smtp-4-1-85-final-sources-1-jar/io/netty/handler/codec/smtp/DefaultSntpResponse.java
- * /opt/cola/permits/1498798484_1670284125.032224/0/netty-codec-smtp-4-1-85-final-sources-1-jar/io/netty/handler/codec/smtp/DefaultLastSntpContent.java
- * /opt/cola/permits/1498798484_1670284125.032224/0/netty-codec-smtp-4-1-85-final-sources-1-jar/io/netty/handler/codec/smtp/package-info.java
- * /opt/cola/permits/1498798484_1670284125.032224/0/netty-codec-smtp-4-1-85-final-sources-1-jar/io/netty/handler/codec/smtp/LastSntpContent.java
- * /opt/cola/permits/1498798484_1670284125.032224/0/netty-codec-smtp-4-1-85-final-sources-1-jar/io/netty/handler/codec/smtp/SntpCommand.java
- * /opt/cola/permits/1498798484_1670284125.032224/0/netty-codec-smtp-4-1-85-final-sources-1-jar/io/netty/handler/codec/smtp/SntpResponse.java

1.125 apache-kafka 2.8.2

1.125.1 Available under license :

DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE

Version 2, December 2004

Copyright (C) 2004 Sam Hocevar <sam@hocevar.net>

Everyone is permitted to copy and distribute verbatim or modified copies of this license document, and changing it is allowed as long as the name is changed.

DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. You just DO WHAT THE FUCK YOU WANT TO.

Zstd-jni: JNI bindings to Zstd Library

Copyright (c) 2015-present, Luben Karavelov/ All rights reserved.

BSD License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing

the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This project bundles some components that are also licensed under the Apache License Version 2.0:

audience-annotations-0.5.0
commons-cli-1.4
commons-lang3-3.8.1
jackson-annotations-2.10.5
jackson-core-2.10.5
jackson-databind-2.10.5.1
jackson-dataformat-csv-2.10.5

jackson-datatype-jdk8-2.10.5
jackson-jaxrs-base-2.10.5
jackson-jaxrs-json-provider-2.10.5
jackson-module-jaxb-annotations-2.10.5
jackson-module-paranamer-2.10.5
jackson-module-scala_2.13-2.10.5
jackson-module-scala_2.12-2.10.5
jakarta.validation-api-2.0.2
javassist-3.27.0-GA
jetty-client-9.4.48.v20220622
jetty-continuation-9.4.48.v20220622
jetty-http-9.4.48.v20220622
jetty-io-9.4.48.v20220622
jetty-security-9.4.48.v20220622
jetty-server-9.4.48.v20220622
jetty-servlet-9.4.48.v20220622
jetty-servlets-9.4.48.v20220622
jetty-util-9.4.48.v20220622
jetty-util-ajax-9.4.48.v20220622
jersey-common-2.34
jersey-server-2.34
log4j-1.2.17
lz4-java-1.7.1
maven-artifact-3.8.1
metrics-core-2.2.0
netty-buffer-4.1.73.Final
netty-codec-4.1.73.Final
netty-common-4.1.73.Final
netty-handler-4.1.73.Final
netty-resolver-4.1.73.Final
netty-transport-4.1.73.Final
netty-transport-native-epoll-4.1.73.Final
netty-transport-native-unix-common-4.1.73.Final
plexus-utils-3.2.1
rocksdbjni-5.18.4
scala-collection-compat_2.13-2.3.0
scala-library-2.13.5
scala-logging_2.13-3.9.2
scala-reflect-2.13.5
scala-java8-compat_2.13-0.9.1
snappy-java-1.1.8.1
zookeeper-3.5.9
zookeeper-jute-3.5.9

=====
This product bundles various third-party components under other open source licenses. This section summarizes those components and their licenses. See licenses/ for text of these licenses.

Eclipse Distribution License - v 1.0
see: licenses/eclipse-distribution-license-1.0

jakarta.activation-api-1.2.1
jakarta.xml.bind-api-2.3.2

Eclipse Public License - v 2.0
see: licenses/eclipse-public-license-2.0

jakarta.annotation-api-1.3.5
jakarta.ws.rs-api-2.1.6
javax.ws.rs-api-2.1.1
hk2-api-2.6.1
hk2-locator-2.6.1
hk2-utils-2.6.1
osgi-resource-locator-1.0.3
aopalliance-repackaged-2.6.1
jakarta.inject-2.6.1
jersey-container-servlet-2.34
jersey-container-servlet-core-2.34
jersey-client-2.34
jersey-hk2-2.34
jersey-media-jaxb-2.31

CDDL 1.1 + GPLv2 with classpath exception
see: licenses/CDDL+GPL-1.1

javax.servlet-api-3.1.0
jaxb-api-2.3.0
activation-1.1.1

MIT License

argparse4j-0.7.0, see: licenses/argparse-MIT
jopt-simple-5.0.4, see: licenses/jopt-simple-MIT
slf4j-api-1.7.30, see: licenses/slf4j-MIT
slf4j-log4j12-1.7.30, see: licenses/slf4j-MIT

BSD 2-Clause

zstd-jni-1.4.9-1, see: licenses/zstd-jni-BSD-2-clause

BSD 3-Clause

jline-3.12.1, see: licenses/jline-BSD-3-clause
paranamer-2.8, see: licenses/paranamer-BSD-3-clause

Do What The F*ck You Want To Public License
see: licenses/DWTFYWTPL

reflections-0.9.12
Copyright (c) 2002-2018, the original author or authors.
All rights reserved.

<https://opensource.org/licenses/BSD-3-Clause>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of JLine nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/*

* Copyright (C) 2011-2017 Tatsuhiro Tsujikawa

*

* Permission is hereby granted, free of charge, to any person
* obtaining a copy of this software and associated documentation
* files (the "Software"), to deal in the Software without
* restriction, including without limitation the rights to use, copy,
* modify, merge, publish, distribute, sublicense, and/or sell copies
* of the Software, and to permit persons to whom the Software is
* furnished to do so, subject to the following conditions:
*
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
* BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
* ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
* CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
* SOFTWARE.

*/

Apache Kafka

Copyright 2022 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

This distribution has a binary dependency on jersey, which is available under the CDDL
License. The source code of jersey can be found at <https://github.com/jersey/jersey/>.

The streams-scala (streams/streams-scala) module was donated by Lightbend and the original code was copyrighted
by them:

Copyright (C) 2018 Lightbend Inc. <<https://www.lightbend.com>>

Copyright (C) 2017-2018 Alexis Seigneurin.

This project contains the following code copied from Apache Hadoop:

clients/src/main/java/org/apache/kafka/common/utils/PureJavaCrc32C.java

Some portions of this file Copyright (c) 2004-2006 Intel Corporation and licensed under the BSD license.

This project contains the following code copied from Apache Hive:

streams/src/main/java/org/apache/kafka/streams/state/internals/Murmur3.java

/*

The MIT License

Copyright (c) 2004-2016 Paul R. Holser, Jr.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including

without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

*/

Copyright (c) 2004-2017 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. **Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

[ParaNamer used to be 'Public Domain', but since it includes a small piece of ASM it is now the same license as that: BSD]

Portions copyright (c) 2006-2018 Paul Hamant & ThoughtWorks Inc
Portions copyright (c) 2000-2007 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.126 x-sync 20220722-snapshot-886fb937

1.126.1 Available under license :

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its

contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.127 jetbrains-annotations 17.0.0

1.127.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2000-2016 JetBrains s.r.o.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1331473649_1653510392.001332/0/1463-annotations-17-0-0-sources-
zip/org/jetbrains/annotations/Contract.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2000-2014 JetBrains s.r.o.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1331473649_1653510392.001332/0/1463-annotations-17-0-0-sources-
zip/org/intellij/lang/annotations/MagicConstant.java
* /opt/cola/permits/1331473649_1653510392.001332/0/1463-annotations-17-0-0-sources-
zip/org/jetbrains/annotations/Nullable.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2000-2015 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1331473649_1653510392.001332/0/1463-annotations-17-0-0-sources-
zip/org/jetbrains/annotations/TestOnly.java
* /opt/cola/permits/1331473649_1653510392.001332/0/1463-annotations-17-0-0-sources-
zip/org/intellij/lang/annotations/Flow.java
* /opt/cola/permits/1331473649_1653510392.001332/0/1463-annotations-17-0-0-sources-
zip/org/jetbrains/annotations/Nls.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2006 Sascha Weinreuter
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at

*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1331473649_1653510392.001332/0/1463-annotations-17-0-0-sources-
zip/org/intellij/lang/annotations/Identifier.java
* /opt/cola/permits/1331473649_1653510392.001332/0/1463-annotations-17-0-0-sources-
zip/org/intellij/lang/annotations/RegExp.java
* /opt/cola/permits/1331473649_1653510392.001332/0/1463-annotations-17-0-0-sources-
zip/org/intellij/lang/annotations/Subst.java
* /opt/cola/permits/1331473649_1653510392.001332/0/1463-annotations-17-0-0-sources-
zip/org/intellij/lang/annotations/Language.java
* /opt/cola/permits/1331473649_1653510392.001332/0/1463-annotations-17-0-0-sources-
zip/org/intellij/lang/annotations/Pattern.java
* /opt/cola/permits/1331473649_1653510392.001332/0/1463-annotations-17-0-0-sources-
zip/org/intellij/lang/annotations/PrintFormat.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2000-2012 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1331473649_1653510392.001332/0/1463-annotations-17-0-0-sources-
zip/org/jetbrains/annotations/NotNull.java
* /opt/cola/permits/1331473649_1653510392.001332/0/1463-annotations-17-0-0-sources-
zip/org/intellij/lang/annotations/JdkConstants.java

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2000-2009 JetBrains s.r.o.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
*/opt/cola/permits/1331473649_1653510392.001332/0/1463-annotations-17-0-0-sources-
zip/org/jetbrains/annotations/PropertyKey.java
*/opt/cola/permits/1331473649_1653510392.001332/0/1463-annotations-17-0-0-sources-
zip/org/jetbrains/annotations/NonNls.java
```

1.128 javapoet 1.13.0

1.128.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2015 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
*/opt/cola/permits/1526005867_1673041436.749772/0/javapoet-1-13-0-sources-3-
jar/com/squareup/javapoet/FieldSpec.java
*/opt/cola/permits/1526005867_1673041436.749772/0/javapoet-1-13-0-sources-3-
```

jar/com/squareup/javapoet/ArrayTypeName.java
* /opt/cola/permits/1526005867_1673041436.749772/0/javapoet-1-13-0-sources-3-
jar/com/squareup/javapoet/MethodSpec.java
* /opt/cola/permits/1526005867_1673041436.749772/0/javapoet-1-13-0-sources-3-
jar/com/squareup/javapoet/NameAllocator.java
* /opt/cola/permits/1526005867_1673041436.749772/0/javapoet-1-13-0-sources-3-
jar/com/squareup/javapoet/ParameterSpec.java
* /opt/cola/permits/1526005867_1673041436.749772/0/javapoet-1-13-0-sources-3-
jar/com/squareup/javapoet/TypeName.java
* /opt/cola/permits/1526005867_1673041436.749772/0/javapoet-1-13-0-sources-3-
jar/com/squareup/javapoet/TypeSpec.java
* /opt/cola/permits/1526005867_1673041436.749772/0/javapoet-1-13-0-sources-3-
jar/com/squareup/javapoet/TypeVariableName.java
* /opt/cola/permits/1526005867_1673041436.749772/0/javapoet-1-13-0-sources-3-
jar/com/squareup/javapoet/WildcardTypeName.java
* /opt/cola/permits/1526005867_1673041436.749772/0/javapoet-1-13-0-sources-3-
jar/com/squareup/javapoet/AnnotationSpec.java
* /opt/cola/permits/1526005867_1673041436.749772/0/javapoet-1-13-0-sources-3-
jar/com/squareup/javapoet/ParameterizedTypeName.java
* /opt/cola/permits/1526005867_1673041436.749772/0/javapoet-1-13-0-sources-3-
jar/com/squareup/javapoet/JavaFile.java
* /opt/cola/permits/1526005867_1673041436.749772/0/javapoet-1-13-0-sources-3-
jar/com/squareup/javapoet/CodeBlock.java
* /opt/cola/permits/1526005867_1673041436.749772/0/javapoet-1-13-0-sources-3-
jar/com/squareup/javapoet/Util.java
* /opt/cola/permits/1526005867_1673041436.749772/0/javapoet-1-13-0-sources-3-
jar/com/squareup/javapoet/CodeWriter.java
No license file was found, but licenses were detected in source scan.

```
/*  
 * Copyright (C) 2014 Google, Inc.  
 *  
 * Licensed under the Apache License, Version 2.0 (the "License");  
 * you may not use this file except in compliance with the License.  
 * You may obtain a copy of the License at  
 *  
 * http://www.apache.org/licenses/LICENSE-2.0  
 *  
 * Unless required by applicable law or agreed to in writing, software  
 * distributed under the License is distributed on an "AS IS" BASIS,  
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
 * See the License for the specific language governing permissions and  
 * limitations under the License.  
 */
```

Found in path(s):

* /opt/cola/permits/1526005867_1673041436.749772/0/javapoet-1-13-0-sources-3-
jar/com/squareup/javapoet/ClassName.java

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2016 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
*/opt/cola/permits/1526005867_1673041436.749772/0/javapoet-1-13-0-sources-3-
jar/com/squareup/javapoet/LineWrapper.java
```

1.129 aws-glue-schema-registry-avro-serializer-deserializer 1.1.9

1.129.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<!--
/*
 * Copyright 2020 Amazon.com, Inc. or its affiliates.
 * Licensed under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
-->
```

Found in path(s):

1.130 commons-compress 1.21

1.130.1 Available under license :

Apache Commons Compress
Copyright 2002-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

The files in the package org.apache.commons.compress.archivers.sevenz were derived from the LZMA SDK, version 9.20 (C/ and CPP/7zip/), which has been placed in the public domain:

"LZMA SDK is placed in the public domain." (<http://www.7-zip.org/sdk.html>)

The test file lbzip2_32767.bz2 has been copied from libbzip2's source repository:

This program, "bzip2", the associated library "libbzip2", and all documentation, are copyright (C) 1996-2019 Julian R Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, jseward@acm.org

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.131 google-guava 30.1-android

1.131.1 Available under license :

Doug Lea

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the

direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.132 kotlin-libraries-bill-of-materials 1.7.20

1.132.1 Available under license :

No license file was found, but licenses were detected in source scan.

<url><http://www.apache.org/licenses/LICENSE-2.0.txt></url>

Found in path(s):

* /opt/cola/permits/1526005977_1673504901.1757238/0/kotlin-bom-1-7-20-pom-zip/kotlin-bom-1.7.20.pom

1.133 netty/transport/classes/epoll

4.1.85.Final

1.133.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Implementation-Title: Netty/Transport/Classes/Epoll

Bundle-Description: Netty is an asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers and clients.

Automatic-Module-Name: io.netty.transport.classes.epoll

Bundle-License: <https://www.apache.org/licenses/LICENSE-2.0>

Bundle-SymbolicName: io.netty.transport-classes-epoll

Implementation-Version: 4.1.85.Final

Built-By: root

Bnd-LastModified: 1668015176512

Bundle-ManifestVersion: 2

Implementation-Vendor-Id: io.netty

Bundle-DocURL: <https://netty.io/>

Bundle-Vendor: The Netty Project

Import-Package: io.netty.buffer;version="[4.1,5)",io.netty.channel,io.netty.channel.socket;version="[4.1,5)",io.netty.channel.unix;version="[4.1,5)",io.netty.util;version="[4.1,5)",io.netty.util.collection;version="[4.1,5)",io.netty.util.concurrent;version="[4.1,5)",io.netty.util.internal;version="[4.1,5)",io.netty.util.internal.logging;version="[4.1,5)",sun.nio.ch;resolution:=optional,org.eclipse.jetty.npn;version="[1,2)";resolution:=optional,org.eclipse.jetty.alpn;version="[1,2)";resolution:=optional

Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.6))"

Tool: Bnd-2.4.1.201501161923

Implementation-Vendor: The Netty Project

Export-Package: io.netty.channel.epoll;uses:="io.netty.buffer,io.netty.channel,io.netty.channel.socket,io.netty.channel.unix,io.netty.util,io.netty.util.concurrent";version="4.1.85"

Bundle-Name: Netty/Transport/Classes/Epoll

Bundle-Version: 4.1.85.Final

Created-By: Apache Maven Bundle Plugin

Build-Jdk: 1.8.0_352

Implementation-URL: <https://netty.io/netty-transport-classes-epoll/>

Found in path(s):

* /opt/cola/permits/1498798605_1670358395.43995/0/netty-transport-classes-epoll-4-1-85-final-jar/META-

INF/MANIFEST.MF

No license file was found, but licenses were detected in source scan.

```
<!--
~ Copyright 2021 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ https://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.
-->
```

Found in path(s):

```
* /opt/cola/permits/1498798605_1670358395.43995/0/netty-transport-classes-epoll-4-1-85-final-jar/META-INF/maven/io.netty/netty-transport-classes-epoll/pom.xml
```

1.134 docker-java 3.2.13

1.134.1 Available under license :

Apache-2.0

1.135 kotlin-scripting-compiler-impl-embeddable 1.7.20

1.135.1 Available under license :

Apache-2.0

1.136 wire 3.7.1

1.136.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.137 golang-glog 20160125-snapshot-23def4e6

1.137.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions,

annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof

in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.138 prometheus-client-model v0.2.0

1.138.1 Available under license :

Data model artifacts for Prometheus.

Copyright 2012-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or

for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason

of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.139 go-logrus v1.8.1

1.139.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Simon Eskildsen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.140 gomega v1.16.0

1.140.1 Available under license :

Copyright (c) 2013-2014 Onsi Fakhouri

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014 Amit Kumar Gupta

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE

LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.141 google-gson 2.8.6

1.141.1 Available under license :

Google Gson

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2008-2011 Google Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.142 jtolds-gls v4.20.0

1.142.1 Available under license :

Copyright (c) 2013, Space Monkey, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS

FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.143 apache-log4j 2.19.0

1.143.1 Available under license :

Apache Log4j Core
Copyright 1999-2012 Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

ResolverUtil.java
Copyright 2005-2006 Tim Fennell

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use,

reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.144 google-uuid v.1.1.2

1.144.1 Available under license :

Paul Borman <borman@google.com>

bmatsuo

shawnps

theory

jboverfelt

dsymonds

cd1

wallclockbuilder

dansouza

Copyright (c) 2009,2014 Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright

notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.145 jetbrains-kotlin-kotlin-stdlib-jdk8 1.4.10

1.145.1 Available under license :

Note that publicsuffixes.gz is compiled from The Public Suffix List:
https://publicsuffix.org/list/public_suffix_list.dat

It is subject to the terms of the Mozilla Public License, v. 2.0:
<https://mozilla.org/MPL/2.0/>

1.146 byte-buddy-agent 1.11.3

1.146.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Copyright `${project.inceptionYear}` - `${current.year}` `${copyright.holder}`

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.147 asm-commons 9.1

1.147.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bundle-DocURL: <http://asm.ow2.org>

Bundle-License: BSD-3-Clause;link=<https://asm.ow2.io/LICENSE.txt>

Bundle-ManifestVersion: 2

Bundle-Name: org.objectweb.asm.commons

Bundle-RequiredExecutionEnvironment: J2SE-1.5

Bundle-SymbolicName: org.objectweb.asm.commons

Bundle-Version: 9.1.0

Export-Package: org.objectweb.asm.commons;version="9.1";uses:="org.obj

ectweb.asm,org.objectweb.asm.signature,org.objectweb.asm.tree"
Implementation-Title: Usefull class adapters based on ASM, a very smal
l and fast Java bytecode manipulation framework
Implementation-Version: 9.1
Import-Package: org.objectweb.asm;version="[9.1,10)",org.objectweb.asm
.signature;version="[9.1,10)",org.objectweb.asm.tree;version="[9.1,10
)"
Module-Requires: org.objectweb.asm;transitive=true,org.objectweb.asm.t
ree;transitive=true,org.objectweb.asm.tree.analysis;transitive=true

Found in path(s):

* /opt/cola/permits/1183889887_1627494130.56/0/asm-commons-9-1-jar/META-INF/MANIFEST.MF

1.148 kotlinpoet 1.7.2

1.148.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use,

reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.149 jna 5.8.0

1.149.1 Available under license :

/*

- * The contents of this file are subject to the Mozilla Public License Version 1.1
- * (the "License"); you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at <http://www.mozilla.org/MPL/>
- *
- * Software distributed under the License is distributed on an "AS IS" basis,
- * WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License
- * for the specific language governing rights and limitations under the License.
- *
- * The Original Code is 'iText, a free JAVA-PDF library'.
- *
- * The Initial Developer of the Original Code is Bruno Lowagie. Portions created by
- * the Initial Developer are Copyright (C) 1999, 2000, 2001, 2002 by Bruno Lowagie.
- * All Rights Reserved.
- * Co-Developer of the code is Paulo Soares. Portions created by the Co-Developer
- * are Copyright (C) 2000, 2001, 2002 by Paulo Soares. All Rights Reserved.

*
* Contributor(s): all the names of the contributors are added in the source code
* where applicable.
*
* Alternatively, the contents of this file may be used under the terms of the
* LGPL license (the "GNU LIBRARY GENERAL PUBLIC LICENSE"), in which case the
* provisions of LGPL are applicable instead of those above. If you wish to
* allow use of your version of this file only under the terms of the LGPL
* License and not to allow others to use your version of this file under
* the MPL, indicate your decision by deleting the provisions above and
* replace them with the notice and other provisions required by the LGPL.
* If you do not delete the provisions above, a recipient may use your version
* of this file under either the MPL or the GNU LIBRARY GENERAL PUBLIC LICENSE.
*
* This library is free software; you can redistribute it and/or modify it
* under the terms of the MPL as stated above or under the terms of the GNU
* Library General Public License as published by the Free Software Foundation;
* either version 2 of the License, or any later version.
*
* This library is distributed in the hope that it will be useful, but WITHOUT
* ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS
* FOR A PARTICULAR PURPOSE. See the GNU Library general Public License for more
* details.
*
* If you didn't download this code from the following link, you should check if
* you aren't using an obsolete version:
* <http://www.lowagie.com/iText/>
*
* This class is generated based on a grammar file provided by SUN, and updated
* by Carsten Hammer. SUN's license agreement can be found at this URL:
* <http://java.sun.com/products/java-media/2D/samples/samples-license.html>
* See also the file sun.txt in directory com.lowagie.text.pdf
*/

Maven Ant Tasks

Copyright 2002-2011 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

=====
=====

Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed their package names from 'org.jfree' to 'clover.org.jfree'. This was necessary to avoid potential name conflicts during instrumentation of a code using the original library when using Clover. No source code of the original library was modified.

=====

=====

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free

software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not

covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If

identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding

machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library

facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by

all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our

decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either

version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
BSD License

Copyright (c) 2000-2006, www.hamcrest.org
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer. Redistributions in binary form must reproduce
the above copyright notice, this list of conditions and the following disclaimer in
the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse
or promote products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
EVENT
SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,

INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache Ant

Copyright 1999-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

The <sync> task is based on code Copyright (c) 2002, Landmark Graphics Corp that has been kindly donated to the Apache Software Foundation.

/*

* Apache License
* Version 2.0, January 2004
* <http://www.apache.org/licenses/>

*

* TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

*

* 1. Definitions.

*

* "License" shall mean the terms and conditions for use, reproduction,
* and distribution as defined by Sections 1 through 9 of this document.

*

* "Licensor" shall mean the copyright owner or entity authorized by
* the copyright owner that is granting the License.

*

* "Legal Entity" shall mean the union of the acting entity and all
* other entities that control, are controlled by, or are under common
* control with that entity. For the purposes of this definition,
* "control" means (i) the power, direct or indirect, to cause the
* direction or management of such entity, whether by contract or
* otherwise, or (ii) ownership of fifty percent (50%) or more of the
* outstanding shares, or (iii) beneficial ownership of such entity.

*

* "You" (or "Your") shall mean an individual or Legal Entity
* exercising permissions granted by this License.

*

* "Source" form shall mean the preferred form for making modifications,
* including but not limited to software source code, documentation
* source, and configuration files.

*

* "Object" form shall mean any form resulting from mechanical

* transformation or translation of a Source form, including but
* not limited to compiled object code, generated documentation,
* and conversions to other media types.

* "Work" shall mean the work of authorship, whether in Source or
* Object form, made available under the License, as indicated by a
* copyright notice that is included in or attached to the work
* (an example is provided in the Appendix below).

* "Derivative Works" shall mean any work, whether in Source or Object
* form, that is based on (or derived from) the Work and for which the
* editorial revisions, annotations, elaborations, or other modifications
* represent, as a whole, an original work of authorship. For the purposes
* of this License, Derivative Works shall not include works that remain
* separable from, or merely link (or bind by name) to the interfaces of,
* the Work and Derivative Works thereof.

* "Contribution" shall mean any work of authorship, including
* the original version of the Work and any modifications or additions
* to that Work or Derivative Works thereof, that is intentionally
* submitted to Licensor for inclusion in the Work by the copyright owner
* or by an individual or Legal Entity authorized to submit on behalf of
* the copyright owner. For the purposes of this definition, "submitted"
* means any form of electronic, verbal, or written communication sent
* to the Licensor or its representatives, including but not limited to
* communication on electronic mailing lists, source code control systems,
* and issue tracking systems that are managed by, or on behalf of, the
* Licensor for the purpose of discussing and improving the Work, but
* excluding communication that is conspicuously marked or otherwise
* designated in writing by the copyright owner as "Not a Contribution."

* "Contributor" shall mean Licensor and any individual or Legal Entity
* on behalf of whom a Contribution has been received by Licensor and
* subsequently incorporated within the Work.

* 2. Grant of Copyright License. Subject to the terms and conditions of
* this License, each Contributor hereby grants to You a perpetual,
* worldwide, non-exclusive, no-charge, royalty-free, irrevocable
* copyright license to reproduce, prepare Derivative Works of,
* publicly display, publicly perform, sublicense, and distribute the
* Work and such Derivative Works in Source or Object form.

* 3. Grant of Patent License. Subject to the terms and conditions of
* this License, each Contributor hereby grants to You a perpetual,
* worldwide, non-exclusive, no-charge, royalty-free, irrevocable
* (except as stated in this section) patent license to make, have made,
* use, offer to sell, sell, import, and otherwise transfer the Work,
* where such license applies only to those patent claims licensable

* by such Contributor that are necessarily infringed by their
* Contribution(s) alone or by combination of their Contribution(s)
* with the Work to which such Contribution(s) was submitted. If You
* institute patent litigation against any entity (including a
* cross-claim or counterclaim in a lawsuit) alleging that the Work
* or a Contribution incorporated within the Work constitutes direct
* or contributory patent infringement, then any patent licenses
* granted to You under this License for that Work shall terminate
* as of the date such litigation is filed.

* 4. Redistribution. You may reproduce and distribute copies of the
* Work or Derivative Works thereof in any medium, with or without
* modifications, and in Source or Object form, provided that You
* meet the following conditions:

* (a) You must give any other recipients of the Work or
* Derivative Works a copy of this License; and

* (b) You must cause any modified files to carry prominent notices
* stating that You changed the files; and

* (c) You must retain, in the Source form of any Derivative Works
* that You distribute, all copyright, patent, trademark, and
* attribution notices from the Source form of the Work,
* excluding those notices that do not pertain to any part of
* the Derivative Works; and

* (d) If the Work includes a "NOTICE" text file as part of its
* distribution, then any Derivative Works that You distribute must
* include a readable copy of the attribution notices contained
* within such NOTICE file, excluding those notices that do not
* pertain to any part of the Derivative Works, in at least one
* of the following places: within a NOTICE text file distributed
* as part of the Derivative Works; within the Source form or
* documentation, if provided along with the Derivative Works; or,
* within a display generated by the Derivative Works, if and
* wherever such third-party notices normally appear. The contents
* of the NOTICE file are for informational purposes only and
* do not modify the License. You may add Your own attribution
* notices within Derivative Works that You distribute, alongside
* or as an addendum to the NOTICE text from the Work, provided
* that such additional attribution notices cannot be construed
* as modifying the License.

* You may add Your own copyright statement to Your modifications and
* may provide additional or different license terms and conditions
* for use, reproduction, or distribution of Your modifications, or
* for any such Derivative Works as a whole, provided Your use,

- * reproduction, and distribution of the Work otherwise complies with
- * the conditions stated in this License.
- *
- * 5. Submission of Contributions. Unless You explicitly state otherwise,
- * any Contribution intentionally submitted for inclusion in the Work
- * by You to the Licensor shall be under the terms and conditions of
- * this License, without any additional terms or conditions.
- * Notwithstanding the above, nothing herein shall supersede or modify
- * the terms of any separate license agreement you may have executed
- * with Licensor regarding such Contributions.
- *
- * 6. Trademarks. This License does not grant permission to use the trade
- * names, trademarks, service marks, or product names of the Licensor,
- * except as required for reasonable and customary use in describing the
- * origin of the Work and reproducing the content of the NOTICE file.
- *
- * 7. Disclaimer of Warranty. Unless required by applicable law or
- * agreed to in writing, Licensor provides the Work (and each
- * Contributor provides its Contributions) on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
- * implied, including, without limitation, any warranties or conditions
- * of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
- * PARTICULAR PURPOSE. You are solely responsible for determining the
- * appropriateness of using or redistributing the Work and assume any
- * risks associated with Your exercise of permissions under this License.
- *
- * 8. Limitation of Liability. In no event and under no legal theory,
- * whether in tort (including negligence), contract, or otherwise,
- * unless required by applicable law (such as deliberate and grossly
- * negligent acts) or agreed to in writing, shall any Contributor be
- * liable to You for damages, including any direct, indirect, special,
- * incidental, or consequential damages of any character arising as a
- * result of this License or out of the use or inability to use the
- * Work (including but not limited to damages for loss of goodwill,
- * work stoppage, computer failure or malfunction, or any and all
- * other commercial damages or losses), even if such Contributor
- * has been advised of the possibility of such damages.
- *
- * 9. Accepting Warranty or Additional Liability. While redistributing
- * the Work or Derivative Works thereof, You may choose to offer,
- * and charge a fee for, acceptance of support, warranty, indemnity,
- * or other liability obligations and/or rights consistent with this
- * License. However, in accepting such obligations, You may act only
- * on Your own behalf and on Your sole responsibility, not on behalf
- * of any other Contributor, and only if You agree to indemnify,
- * defend, and hold each Contributor harmless for any liability
- * incurred by, or claims asserted against, such Contributor by reason
- * of your accepting any such warranty or additional liability.

*
* END OF TERMS AND CONDITIONS
*
* APPENDIX: How to apply the Apache License to your work.
*
* To apply the Apache License to your work, attach the following
* boilerplate notice, with the fields enclosed by brackets "[]"
* replaced with your own identifying information. (Don't include
* the brackets!) The text should be enclosed in the appropriate
* comment syntax for the file format. We also recommend that a
* file or class name and description of purpose be included on the
* same "printed page" as the copyright notice for easier
* identification within third-party archives.
*
* Copyright [yyyy] [name of copyright owner]
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

W3C SOFTWARE NOTICE AND LICENSE

<http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body

of any redistributed or derivative code.

3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". Otherwise, this version is the same as the previous version and is written so as to preserve the Free Software Foundation's assessment of GPL compatibility and OSI's certification under the Open Source Definition. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.

Joseph Reagle <site-policy@w3.org>

This license came from: <http://www.megginson.com/SAX/copying.html>
However please note future versions of SAX may be covered
under <http://saxproject.org/?selected=pd>

SAX2 is Free!

I hereby abandon any property rights to SAX 2.0 (the Simple API for XML), and release all of the SAX 2.0 source code, compiled code, and documentation contained in this distribution into the Public Domain. SAX comes with NO WARRANTY or guarantee of fitness for any purpose.

David Megginson, david@megginson.com

2000-05-05

=====
=====
Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed their package names from 'com.keypoint/org.jfree' to 'clover.com.keypoint/clover.org.jfree'. This was necessary to avoid potential name conflicts during instrumentation of a code using the original library when using Clover. No source code of the original library was modified.

=====
=====

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these

rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages

are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated

straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of

its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form

under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies

the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and

conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is

safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a

result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Commons Lang

Copyright 2001-2011 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (<http://www.apache.org/>).

/*

* The Apache Software License, Version 1.1

*

* Copyright (c) 2000-2003 The Apache Software Foundation. All rights reserved.

*

* Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

*

* 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

*

* 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

*

* 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgement:

* "This product includes software developed by the

* Apache Software Foundation (<http://www.apache.org/>)."

* Alternately, this acknowledgement may appear in the software itself, if and wherever such third-party acknowledgements normally appear.

*

* 4. The names "Ant" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.

*

* 5. Products derived from this software may not be called "Apache" nor may "Apache" appear in their names without prior written permission of the Apache Group.

*

* THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* =====

*

* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation. For more
* information on the Apache Software Foundation, please see
* <<http://www.apache.org/>>.
*/

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

- a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
- b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

- a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

- a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the Combined Work with a copy of the GNU GPL and this license

document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

MOZILLA PUBLIC LICENSE

Version 1.1

1. Definitions.

1.0.1. "Commercial Use" means distribution or otherwise making the Covered Code available to a third party.

1.1. "Contributor" means each entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

1.4. "Electronic Distribution Mechanism" means a mechanism generally

accepted in the software development community for the electronic transfer of data.

1.5. "Executable" means Covered Code in any form other than Source Code.

1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. Source Code License.

2.1. The Initial Developer Grant.

The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and

(b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

(c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

2.2. Contributor Grant.

Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Application of License.

The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code.

Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification

has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications.

You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters

(a) Third Party Claims.

If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

(b) Contributor APIs.

If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

(c) Representations.

Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

3.5. Required Notices.

You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely

to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions.

You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.7. Larger Works.

You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to

statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Application of this License.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

6. Versions of the License.

6.1. New Versions.

Netscape Communications Corporation ("Netscape") may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. Effect of New Versions.

Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

6.3. Derivative Works.

If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrases "Mozilla", "MOZILLAPL", "MOZPL", "Netscape", "MPL", "NPL" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF

DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:

(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as

by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS.

The Covered Code is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

11. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United

States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

12. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

13. MULTIPLE-LICENSED CODE.

Initial Developer may designate portions of the Covered Code as "Multiple-Licensed". "Multiple-Licensed" means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the NPL or the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

EXHIBIT A -Mozilla Public License.

``The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.mozilla.org/MPL/>

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is _____.

The Initial Developer of the Original Code is _____.
Portions created by _____ are Copyright (C) _____
_____. All Rights Reserved.

Contributor(s): _____.

Alternatively, the contents of this file may be used under the terms of the _____ license (the "[_____] License"), in which case the provisions of [_____] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [_____] License and not to allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [_____] License. If you do not delete the provisions above, a recipient may use your version of this file under either the MPL or the [_____] License."

[NOTE: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original Code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]

/*

Copyright (c) 2000, Derek Petillo
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Praxis Software nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

Public Domain Dedication

This license is acceptable for Free Cultural Works.

Copyright-Only Dedication (based on United States law) or Public Domain Certification

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

Open Source License for the overlibmws Package

1. License coverage

Note that this license only covers the script library (javascript core and plugin modules) and not any supporting material such as the overlibmws website or its online documentation and support files. You may not reproduce the website or its online material without explicit written permission from the author, but can freely incorporate scripts and procedures which are demonstrated in that material into your own HTML or XML documents.

2. License (Artistic)

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.

"Copyright Holder" is whoever is named in the copyright or copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.

You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

- place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

- use the modified Package only within your corporation or organization.

- rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

- make other distribution arrangements with the Copyright Holder.

You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

- distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

- accompany the distribution with the machine-readable source of the Package with your modifications.

- accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names, and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

- make other distribution arrangements with the Copyright Holder.

You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.

The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.

C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package.

The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND

FITNESS FOR A PARTICULAR PURPOSE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

=====
=====

Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed their package names from 'it.unimi.dsi.fastutil' to 'clover.it.unimi.dsi.fastutil'. This was necessary to avoid potential name conflicts during instrumentation of a code using the original library when using Clover. No source code of the original library was modified.

=====
=====

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling
it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the
library, and (2) we offer you this license, which gives you legal
permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in

non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does

and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based

on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and

therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License.

Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the

user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any

particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU

Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
Javolution - Java(TM) Solution for Real-Time and Embedded Systems
Copyright (c) 2006, Javolution (<http://javolution.org>)
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The libffi source distribution contains certain code that is not part of libffi, and is only used as tooling to assist with the building and testing of libffi. This includes the msvcc.sh script used to wrap the Microsoft compiler with GNU compatible command-line options, make_sunver.pl, and the libffi test code distributed in the testsuite/libffi.bhaible directory. This code is distributed with

libffi for the purpose of convenience only, and libffi is in no way derived from this code.

msvcc.sh an testsuite/libffi.bhaible are both distributed under the terms of the GNU GPL version 2, as below.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy,

distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the

Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding

those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License.

libffi - Copyright (c) 1996-2020 Anthony Green, Red Hat, Inc and others.
See source files for details.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache Commons BeanUtils
Copyright 2000-2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

=====
=====

Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed
their package names from 'org.apache.velocity' to 'clover.org.apache.velocity'. This was necessary to
avoid potential name conflicts during instrumentation of a code using the original library when using Clover.
No source code of the original library was modified.

=====
=====

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each

Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

/*

\$Id: LICENSE.txt,v 1.1.1.1 2004/07/01 13:59:13 jvanzyl Exp \$

Copyright 2002 (C) The Codehaus. All Rights Reserved.

Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain copyright statements and notices. Redistributions must also contain a copy of this document.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name "classworlds" must not be used to endorse or promote products derived from this Software without prior written permission of The Codehaus. For written permission, please contact bob@codehaus.org.
4. Products derived from this Software may not be called "classworlds" nor may "classworlds" appear in their names without prior written permission of The Codehaus. "classworlds" is a registered trademark of The Codehaus.
5. Due credit should be given to The Codehaus.
(<http://classworlds.codehaus.org/>).

THIS SOFTWARE IS PROVIDED BY THE CODEHAUS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE CODEHAUS OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES

(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

=====
=====
Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed their package names from the 'antlr.*' to the 'clover.antlr.*'. This was necessary to avoid potential name conflicts during instrumentation of a code using the original library when using Clover.
No source code of the original library was modified.

=====
=====
SOFTWARE RIGHTS

ANTLR 1989-2006 Developed by Terence Parr
Partially supported by University of San Francisco & jGuru.com

We reserve no legal rights to the ANTLR--it is fully in the public domain. An individual or company may do whatever they wish with source code distributed with ANTLR or the code generated by ANTLR, including the incorporation of ANTLR, or its output, into commercial software.

We encourage users to develop software with ANTLR. However, we do ask that credit is given to us for developing ANTLR. By "credit", we mean that if you use ANTLR or incorporate any source code into one of your programs (commercial product, research project, or otherwise) that you acknowledge this fact somewhere in the documentation, research report, etc... If you like ANTLR and have developed a nice tool with the output, please mention that you developed it using ANTLR. In addition, we ask that the headers remain intact in our source code. As long as these guidelines are kept, we expect to continue enhancing this system and expect to make other tools available as they are completed.

The primary ANTLR guy:

Terence Parr
parrt@cs.usfca.edu

parrr@antlr.org

=====
=====
Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed their package name from 'com.google.common' to 'clover.com.google.common'. This was necessary to avoid potential name conflicts during instrumentation of a code using the original library when using Clover. No source code of the original library was modified.

=====
=====

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct

or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of

this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following

boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

=====
=====

Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed their package name from 'com.google.json' to 'clover.com.google.json'. This was necessary to avoid potential name conflicts during instrumentation of a code using the original library when using Clover. No source code of the original library was modified.

=====
=====

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

=====
=====

Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed their package name from 'com.lowagie' to 'clover.com.lowagie'. This was necessary to avoid potential name conflicts during instrumentation of a code using the original library when using Clover.

No source code of the original library was modified.

=====

MOZILLA PUBLIC LICENSE

Version 1.1

1. Definitions.

1.0.1. "Commercial Use" means distribution or otherwise making the Covered Code available to a third party.

1.1. "Contributor" means each entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

1.4. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. "Executable" means Covered Code in any form other than Source Code.

1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a

Modification is:

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. Source Code License.

2.1. The Initial Developer Grant.

The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

(c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

2.2. Contributor Grant.

Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by

that Contributor.

3. Distribution Obligations.

3.1. Application of License.

The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code.

Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications.

You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters

(a) Third Party Claims.

If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will

know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

(b) Contributor APIs.

If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

(c) Representations.

Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

3.5. Required Notices.

You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions.

You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included

in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.7. Larger Works.

You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Application of this License.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

6. Versions of the License.

6.1. New Versions.

Netscape Communications Corporation ("Netscape") may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. Effect of New Versions.

Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

6.3. Derivative Works.

If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrases "Mozilla", "MOZILLAPL", "MOZPL", "Netscape", "MPL", "NPL" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:

(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF

LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS.

The Covered Code is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

11. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

12. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

13. MULTIPLE-LICENSED CODE.

Initial Developer may designate portions of the Covered Code as "Multiple-Licensed". "Multiple-Licensed" means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the NPL or the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

EXHIBIT A -Mozilla Public License.

``The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.mozilla.org/MPL/>

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is _____.

The Initial Developer of the Original Code is _____.
Portions created by _____ are Copyright (C) _____
_____. All Rights Reserved.

Contributor(s): _____.

Alternatively, the contents of this file may be used under the terms of the _____ license (the "[] License"), in which case the provisions of [] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [] License and not to allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [] License. If you do not delete the provisions above, a recipient may use your version of this file under either the MPL or the [] License."

[NOTE: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original Code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]

Apache Velocity

Copyright (C) 2000-2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

onoMnApeShTpQtDJbcUgJTIFONPQeUndIgfQWWNNddIwBl
mi2Kp5RjfhIJdGCSO<bOTNof2KNxm9KCi5lxEyKI9BJW3p
qOPQUXpopOopMMPqnPnXXQPNOPNRnqQNQqStwVxuQSTtVW
UrwSUSSTVwxWSXNmrrpnmqmUUnpsvpntsmmmmmUUnpsvp
ntsmmmmmUUFmbkWJlroZbW4bsbilmjbkqUUnmmmm
Copyright 2001-2005 (C) MetaStuff, Ltd. All Rights Reserved.

Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain copyright statements and notices. Redistributions must also contain a copy of this document.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name "DOM4J" must not be used to endorse or promote products derived from this Software without prior written permission of MetaStuff, Ltd. For written permission, please contact dom4j-info@metastuff.com.
4. Products derived from this Software may not be called "DOM4J" nor may "DOM4J" appear in their names without prior written permission of MetaStuff, Ltd. DOM4J is a registered trademark of MetaStuff, Ltd.
5. Due credit should be given to the DOM4J Project - <http://www.dom4j.org>

THIS SOFTWARE IS PROVIDED BY METASTUFF, LTD. AND CONTRIBUTORS ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL METASTUFF, LTD. OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/*--

\$Id: LICENSE.txt,v 1.11 2004/02/06 09:32:57 jhunter Exp \$

Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.
3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact <request_AT_jdom_DOT_org>.
4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following:

"This product includes software developed by the
JDOM Project (<http://www.jdom.org/>)."

Alternatively, the acknowledgment may be graphical using the logos available at <http://www.jdom.org/images/logos>.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the JDOM Project and was originally created by Jason Hunter <jhunter_AT_jdom_DOT_org> and

Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information on the JDOM Project, please see <<http://www.jdom.org/>>.

*/

From: <http://www.json.org/license.html>

=====
=====
Copyright (c) 2002 JSON.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The Software shall be used for Good, not Evil.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

SPDX-License-Identifier: Apache-2.0 OR LGPL-2.1

Java Native Access (JNA) is licensed under the LGPL, version 2.1 or later, or (from version 4.0 onward) the Apache License, version 2.0.

You can freely decide which license you want to apply to the project.

You may obtain a copy of the LGPL License at:

<http://www.gnu.org/licenses/licenses.html>

A copy is also included in the downloadable source code package containing JNA, in file "LGPL2.1", under the same directory as this file.

You may obtain a copy of the Apache License at:

<http://www.apache.org/licenses/>

A copy is also included in the downloadable source code package containing JNA, in file "AL2.0", under the same directory as this file.

Commercial support may be available, please e-mail twall[at]users[dot]sf[dot]net.

=====
=====
Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed their package name from 'org.apache.commons' to 'clover.org.apache.commons'. This was necessary to avoid potential name conflicts during instrumentation of a code using the original library when using Clover. No source code of the original library was modified.

=====
=====
Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but

not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2005 - 2009 Taras Puchko
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS

INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/******

* CruiseControl, a Continuous Integration Toolkit
* Copyright (c) 2001-2003, ThoughtWorks, Inc.
* 651 W Washington Ave. Suite 500
* Chicago, IL 60661 USA
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* + Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
*
* + Redistributions in binary form must reproduce the above
* copyright notice, this list of conditions and the following
* disclaimer in the documentation and/or other materials provided
* with the distribution.
*
* + Neither the name of ThoughtWorks, Inc., CruiseControl, nor the
* names of its contributors may be used to endorse or promote
* products derived from this software without specific prior
* written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
* PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
* LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
* NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*****/

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

http://www.atlassian.com/dms/wac/Atlassian_EULA_4-2.pdf

GNU LIBRARY GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is
numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some
specially designated Free Software Foundation software, and to any
other libraries whose authors decide to use it. You can use it for
your libraries, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if
you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link a program with the library, you must provide
complete object files to the recipients so that they can relink them
with the library, after making changes to the library and recompiling
it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright

the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact

all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest

your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License.

Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies,

or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU

FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

/*--

\$Id: LICENSE.txt,v 1.11 2004/02/06 09:32:57 jhunter Exp \$

Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.
3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact <request_AT_jdom_DOT_org>.
4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following:

"This product includes software developed by the
JDOM Project (<http://www.jdom.org/>)."

Alternatively, the acknowledgment may be graphical using the logos available at <http://www.jdom.org/images/logos>.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT

CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the JDOM Project and was originally created by Jason Hunter <jhunter_AT_jdom_DOT_org> and Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information on the JDOM Project, please see <<http://www.jdom.org/>>.

*/

Indiana University Extreme! Lab Software License

Version 1.1.1

Copyright (c) 2002 Extreme! Lab, Indiana University. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

"This product includes software developed by the Indiana University Extreme! Lab (<http://www.extreme.indiana.edu/>)."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Indiana Univeristy" and "Indiana Univeristy Extreme! Lab" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact <http://www.extreme.indiana.edu/>.

5. Products derived from this software may not use "Indiana Univeristy"

name nor may "Indiana Univeristy" appear in their name, without prior written permission of the Indiana University.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS, COPYRIGHT HOLDERS OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====
=====
Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed their package names from the 'org.apache.commons' to the 'clover.org.apache.commons'. This was necessary to avoid potential name conflicts during instrumentation of a code using the original library when using Clover. No source code of the original library was modified.

=====
=====
Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

/*

File: Core.js

Description:

Provides common utility functions and the Class object used internally by the library.

Also provides the <TreeUtil> object for manipulating JSON tree structures

Some of the Basic utility functions and the Class system are based in the MooTools Framework <<http://mootools.net>>. Copyright (c) 2006-2009 Valerio Proietti, <<http://mad4milk.net/>>. MIT license <<http://mootools.net/license.txt>>.

Author:

Nicolas Garcia Belmonte

Copyright:

Copyright 2008-2009 by Nicolas Garcia Belmonte.

Homepage:

<<http://thejit.org>>

Version:

1.1.2

License:

BSD License

- > Redistribution and use in source and binary forms, with or without
- > modification, are permitted provided that the following conditions are met:
- > * Redistributions of source code must retain the above copyright
- > notice, this list of conditions and the following disclaimer.
- > * Redistributions in binary form must reproduce the above copyright
- > notice, this list of conditions and the following disclaimer in the
- > documentation and/or other materials provided with the distribution.
- > * Neither the name of the organization nor the
- > names of its contributors may be used to endorse or promote products
- > derived from this software without specific prior written permission.
- >
- > THIS SOFTWARE IS PROVIDED BY Nicolas Garcia Belmonte ``AS IS" AND ANY
- > EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
- > WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
- > DISCLAIMED. IN NO EVENT SHALL Nicolas Garcia Belmonte BE LIABLE FOR ANY
- > DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
- > (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
- > LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
- > ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
- > (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
- > SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
- */

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.150 kotlinox-serialization-core 1.4.0

1.150.1 Available under license :

Apache-2.0

1.151 aws-java-sdk-::-http-clients-::-url-connection 2.17.122

1.151.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but

not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by
Amazon Technologies, Inc (<http://www.amazon.com/>).

THIRD PARTY COMPONENTS

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:
Apache Commons Lang
Copyright 2001-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

1.152 vmihailenco/msgpack v4.0.4

1.152.1 Available under license :

Copyright (c) 2013 The github.com/vmihailenco/msgpack Authors.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.153 JetBrains-Kotlin-Kotlin-Scripting-Compiler-Embeddable 1.7.20

1.153.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

```
* Copyright 2010-2017 JetBrains s.r.o.  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

Found in path(s):

```
* /opt/cola/permits/1526005771_1673461421.0170622/0/kotlin-scripting-compiler-embeddable-1-7-20-sources-  
jar/org/jetbrains/kotlin/scripting/compiler/plugin/repl/DelegatePackageMemberDeclarationProvider.kt  
No license file was found, but licenses were detected in source scan.
```

```
/*  
* Copyright 2010-2015 JetBrains s.r.o.  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

Found in path(s):

```
* /opt/cola/permits/1526005771_1673461421.0170622/0/kotlin-scripting-compiler-embeddable-1-7-20-sources-  
jar/org/jetbrains/kotlin/scripting/compiler/plugin/repl/messages/DiagnosticMessageHolder.kt
```

1.154 netty/transport/classes/kqueue

4.1.85.Final

1.154.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Implementation-Title: Netty/Transport/Classes/KQueue

Bundle-Description: Netty is an asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers and clients.

Automatic-Module-Name: io.netty.transport.classes.kqueue

Bundle-License: <https://www.apache.org/licenses/LICENSE-2.0>

Bundle-SymbolicName: io.netty.transport-classes-kqueue

Implementation-Version: 4.1.85.Final

Built-By: chris

Bnd-LastModified: 1668019095692

Bundle-ManifestVersion: 2

Implementation-Vendor-Id: io.netty

Bundle-DocURL: <https://netty.io/>

Bundle-Vendor: The Netty Project

Import-Package: io.netty.buffer;version="[4.1,5)",io.netty.channel,io.netty.channel.socket;version="[4.1,5)",io.netty.channel.unix;version="[4.1,5)",io.netty.util;version="[4.1,5)",io.netty.util.collection;version="[4.1,5)",io.netty.util.concurrent;version="[4.1,5)",io.netty.util.internal;version="[4.1,5)",io.netty.util.internal.logging;version="[4.1,5)",sun.nio.ch;resolution:=optional,org.eclipse.jetty.npn;version="[1,2)";resolution:=optional,org.eclipse.jetty.alpn;version="[1,2)";resolution:=optional

Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.6))"

Tool: Bnd-2.4.1.201501161923

Implementation-Vendor: The Netty Project

Export-Package: io.netty.channel.kqueue;uses="io.netty.buffer,io.netty.channel,io.netty.channel.socket,io.netty.channel.unix,io.netty.util.concurrent";version="4.1.85"

Bundle-Name: Netty/Transport/Classes/KQueue

Bundle-Version: 4.1.85.Final

Created-By: Apache Maven Bundle Plugin

Build-Jdk: 1.8.0_312

Implementation-URL: <https://netty.io/netty-transport-classes-kqueue/>

Found in path(s):

* /opt/cola/permits/1498798404_1670356710.0250175/0/netty-transport-classes-kqueue-4-1-85-final-jar/META-INF/MANIFEST.MF

No license file was found, but licenses were detected in source scan.

<!--

~ Copyright 2021 The Netty Project

~

~ The Netty Project licenses this file to you under the Apache License,

~ version 2.0 (the "License"); you may not use this file except in compliance

~ with the License. You may obtain a copy of the License at:
~
~ <https://www.apache.org/licenses/LICENSE-2.0>
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.
-->

Found in path(s):

* /opt/cola/permits/1498798404_1670356710.0250175/0/netty-transport-classes-kqueue-4-1-85-final-jar/META-INF/maven/io.netty/netty-transport-classes-kqueue/pom.xml

1.155 snappy-java 1.1.8.1

1.155.1 Available under license :

This product includes software developed by Google
Snappy: <http://code.google.com/p/snappy/> (New BSD License)

This product includes software developed by Apache
PureJavaCrc32C from apache-hadoop-common <http://hadoop.apache.org/>
(Apache 2.0 license)

This library contained statically linked libstdc++. This inclusion is allowed by
"GCC Runtime Library Exception"
<http://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html>

== Contributors ==

- * Tatu Saloranta
- * Providing benchmark suite
- * Alec Wysoker
- * Performance and memory usage improvement

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.156 kotlin-stdlib-common 1.7.20

1.156.1 Available under license :

Apache-2.0

1.157 junit 4.12

1.157.1 Available under license :

JUnit

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

- a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
- b) in the case of each subsequent Contributor:
 - i) changes to the Program, and
 - ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability

and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

1.158 error_prone_annotations 2.3.4

1.158.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2014 The Error Prone Authors.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.

*/

Found in path(s):

- * /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/concurrent/UnlockMethod.java
- * /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/concurrent/LockMethod.java
- * /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/NoAllocation.java

No license file was found, but licenses were detected in source scan.

/*

- * Copyright 2016 The Error Prone Authors.
- *
- * Licensed under the Apache License, Version 2.0 (the "License");
- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.

*/

Found in path(s):

- * /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/RestrictedApi.java
- * /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/DoNotMock.java
- * /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/CompatibleWith.java
- * /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/FormatMethod.java
- * /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/FormatString.java
- * /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/MustBeClosed.java

No license file was found, but licenses were detected in source scan.

/*

- * Copyright 2017 The Error Prone Authors.

*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/CheckReturnValue.java
* /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/concurrent/GuardedBy.java
* /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/OverridingMethodsMustInvokeSuper.java
* /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/DoNotCall.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2015 The Error Prone Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/SuppressPackageLocation.java
* /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/ForOverride.java
* /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/IncompatibleModifiers.java

* /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/concurrent/LazyInit.java
* /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/RequiredModifiers.java
* /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/Var.java
* /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/CanIgnoreReturnValue.java
* /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/CompileTimeConstant.java
* /opt/cola/permits/1206718612_1632455182.54/0/error-prone-annotations-2-3-4-sources-9-jar/com/google/errorprone/annotations/Immutable.java

1.159 wire-protocol-buffer-compiler 3.7.1

1.159.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use,

reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.160 burntsushi-toml v0.3.1

1.160.1 Available under license :

The MIT License (MIT)

Copyright (c) 2013 TOML authors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.161 apache-commons-collections 3.2.2

1.161.1 Available under license :

Apache Commons Collections

Copyright 2001-2015 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.162 aws-java-sdk-::-services-::-amazon-cloudwatch 2.17.101

1.162.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by
Amazon Technologies, Inc (<http://www.amazon.com/>).

THIRD PARTY COMPONENTS

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:
Apache Commons Lang
Copyright 2001-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

1.163 x-net v0.1.0

1.163.1 Available under license :

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.164 allegro-bigcache v2.2.5

1.164.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.165 go-check-check 20190902-snapshot-41f04d3b

1.165.1 Available under license :

Gocheck - A rich testing framework for Go

Copyright (c) 2010-2013 Gustavo Niemeyer <gustavo@niemeyer.net>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR

ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.166 project-lombok 1.18.12

1.166.1 Available under license :

Copyright (C) 2009-2015 The Project Lombok Authors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.167 aws-java-sdk-:::-metrics-interface

2.17.122

1.167.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by Amazon Technologies, Inc (<http://www.amazon.com/>).

THIRD PARTY COMPONENTS

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:
Apache Commons Lang
Copyright 2001-2020 The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (<https://www.apache.org/>).

1.168 dgryski/go-rendezvous 20200823-snapshot-9f7001d1

1.168.1 Available under license :

The MIT License (MIT)

Copyright (c) 2017-2020 Damian Gryski <damian@gryski.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in

all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.169 dgraph-io/ristretto v0.1.0

1.169.1 Available under license :

bbloom.go

// The MIT License (MIT)

// Copyright (c) 2014 Andreas Briebe, eduToolbox@Bri-C GmbH, Sarstedt

// Permission is hereby granted, free of charge, to any person obtaining a copy of
// this software and associated documentation files (the "Software"), to deal in
// the Software without restriction, including without limitation the rights to
// use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
// the Software, and to permit persons to whom the Software is furnished to do so,
// subject to the following conditions:

// The above copyright notice and this permission notice shall be included in all
// copies or substantial portions of the Software.

// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
// IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
// FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
// COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
// IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
// CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

rtutil.go

// MIT License

// Copyright (c) 2019 Ewan Chou

// Permission is hereby granted, free of charge, to any person obtaining a copy
// of this software and associated documentation files (the "Software"), to deal
// in the Software without restriction, including without limitation the rights
// to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
// copies of the Software, and to permit persons to whom the Software is
// furnished to do so, subject to the following conditions:

// The above copyright notice and this permission notice shall be included in all
// copies or substantial portions of the Software.

// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
// IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
// FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
// AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
// LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
// OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
// SOFTWARE.

Modifications:

/*

* Copyright 2019 Dgraph Labs, Inc. and Contributors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.170 reactive-streams v1.0.3

1.170.1 Available under license :

Copyright Statement for Contributions to the Reactive Streams Project

=====

I hereby represent that all present, past and future contributions I make to the Reactive Streams project (which includes all repositories owned by the reactive-streams github organization) are governed by the Creative Commons Zero 1.0 Universal copyright statement, placing my contributions in the public domain. This entails that to the extent possible under law I waive all copyright and related or neighboring rights to the code or documents I contribute. I also represent that I have the authority to perform the above waiver with respect to the entirety of my contributions.

The text of the copyright statement is included in the COPYING file at the root of the reactive-streams repository at <https://github.com/reactive-streams/reactive-streams-jvm/blob/master/COPYING>.

Underwriting parties:

github name | Real Name, Email Address used for git commits, Company

-----+

rkuhn | Roland Kuhn, rk@rkuhn.info, Typesafe Inc.
benjchristensen| Ben Christensen, benjchristensen@gmail.com, Netflix Inc.
viktorklang | Viktor Klang, viktorklang@gmail.com, Typesafe Inc.
smaldini | Stephane Maldini, stephane.maldini@gmail.com, Pivotal Software Inc.
savulchik | Stanislav Savulchik, s.savulchik@gmail.com
ktoso | Konrad Malawski, konrad.malawski@project13.pl, Typesafe Inc.
ouertani | Slim Ouertani, ouertani@gmail.com
2m | Martynas Mickevicius, mmartynas@gmail.com, Typesafe Inc.
ldaley | Luke Daley, luke.daley@gradleware.com, Gradleware Inc.
colinrgodsey | Colin Godsey, crgodsey@gmail.com, MediaMath Inc.

davidmoten | Dave Moten, davidmoten@gmail.com
briantopping | Brian Topping, brian.topping@gmail.com, Mauswerks LLC
rstoyanchev | Rossen Stoyanchev, rstoyanchev@pivotal.io, Pivotal
BjornHamels | Bjrn Hamels, bjorn@hamels.nl
JakeWharton | Jake Wharton, jakewharton@gmail.com
anthonyvdotbe | Anthony Vanelverdinghe, anthonyv.be@outlook.com
seratch | Kazuhiro Sera, seratch@gmail.com, SmartNews, Inc.
akarnokd | David Karnok, akarnokd@gmail.com
egetman | Evgeniy Getman, getman.eugene@gmail.com
patriknw | Patrik Nordwall, patrik.nordwall@gmail.com, Lightbend Inc
angelsanz | ngel Sanz, angelsanz@users.noreply.github.com
shenghaiyang | , shenghaiyang@aliyun.com
kiiadi | Kyle Thomson, kylthoms@amazon.com, Amazon.com
jroper | James Roper, james@jazzy.id.au, Lightbend Inc.
olegdokuka | Oleh Dokuka, shadowgun@.i.ua, Netifi Inc.
Scottmitch | Scott Mitchell, scott_mitchell@apple.com, Apple Inc.
retronym | Jason Zaugg, jzaugg@gmail.com, Lightbend Inc.
Creative Commons Legal Code

CC0 1.0 Universal

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS DOCUMENT DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER.

Statement of Purpose

The laws of most jurisdictions throughout the world automatically confer exclusive Copyright and Related Rights (defined below) upon the creator and subsequent owner(s) (each and all, an "owner") of an original work of authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work for the purpose of contributing to a commons of creative, cultural and scientific works ("Commons") that the public can reliably and without fear of later claims of infringement build upon, modify, incorporate in other works, reuse and redistribute as freely as possible in any form whatsoever and for any purposes, including without limitation commercial purposes. These owners may contribute to the Commons to promote the ideal of a free culture and the further production of creative, cultural and scientific works, or to gain reputation or greater distribution for their Work in part through the use and efforts of others.

For these and/or other purposes and motivations, and without any expectation of additional consideration or compensation, the person associating CC0 with a Work (the "Affirmer"), to the extent that he or she is an owner of Copyright and Related Rights in the Work, voluntarily elects to apply CC0 to the Work and publicly distribute the Work under its terms, with knowledge of his or her Copyright and Related Rights in the Work and the meaning and intended legal effect of CC0 on those rights.

1. Copyright and Related Rights. A Work made available under CC0 may be protected by copyright and related or neighboring rights ("Copyright and Related Rights"). Copyright and Related Rights include, but are not limited to, the following:

- i. the right to reproduce, adapt, distribute, perform, display, communicate, and translate a Work;
- ii. moral rights retained by the original author(s) and/or performer(s);
- iii. publicity and privacy rights pertaining to a person's image or likeness depicted in a Work;
- iv. rights protecting against unfair competition in regards to a Work, subject to the limitations in paragraph 4(a), below;
- v. rights protecting the extraction, dissemination, use and reuse of data in a Work;
- vi. database rights (such as those arising under Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, and under any national implementation thereof, including any amended or successor version of such directive); and
- vii. other similar, equivalent or corresponding rights throughout the world based on applicable law or treaty, and any national implementations thereof.

2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and successors, fully intending that such Waiver shall not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public as contemplated by Affirmer's express Statement of Purpose.

3. Public License Fallback. Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the Waiver shall be preserved to the maximum extent permitted taking into account Affirmer's express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "License"). The License shall be deemed effective as of the date CC0 was applied by Affirmer to the Work. Should any part of the License for any reason be judged legally invalid or ineffective under applicable law, such partial invalidity or ineffectiveness shall not invalidate the remainder of the License, and in such case Affirmer hereby affirms that he or she will not (i) exercise any of his or her remaining Copyright and Related Rights in the Work or (ii) assert any associated claims and causes of action with respect to the Work, in either case contrary to Affirmer's express Statement of Purpose.

4. Limitations and Disclaimers.

- a. No trademark or patent rights held by Affirmer are waived, abandoned, surrendered, licensed or otherwise affected by this document.
- b. Affirmer offers the Work as-is and makes no representations or warranties of any kind concerning the Work, express, implied, statutory or otherwise, including without limitation warranties of title, merchantability, fitness for a particular purpose, non infringement, or the absence of latent or other defects, accuracy, or the present or absence of errors, whether or not discoverable, all to the greatest extent permissible under applicable law.
- c. Affirmer disclaims responsibility for clearing rights of other persons that may apply to the Work or any use thereof, including without limitation any person's Copyright and Related Rights in the Work. Further, Affirmer disclaims responsibility for obtaining any necessary consents, permissions or other rights required for any use of the Work.
- d. Affirmer understands and acknowledges that Creative Commons is not a party to this document and has no duty or obligation with respect to this CC0 or use of the Work.

Licensed under Public Domain (CC0)

To the extent possible under law, the person who associated CC0 with this code has waived all copyright and related or neighboring rights to this code.

You should have received a copy of the CC0 legalcode along with this work. If not, see <<http://creativecommons.org/publicdomain/zero/1.0/>>.

1.171 netty-transport-native-unix-common

4.1.85.Final

1.171.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2018 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
```

Found in path(s):

```
*/opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/Buffer.java
*/opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/netty_unix_buffer.h
*/opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/PreferredDirectByteBufferAllocator.java
*/opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/netty_unix_buffer.c
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
```

* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):

* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/package-info.java
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/UnixChannelOption.java
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/IovArray.java
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/Unix.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2016 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* <https://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):

* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/ErrorsStaticallyReferencedJniMethods.java
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/netty_unix_util.c
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/netty_unix_limits.c
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/netty_unix_limits.h
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/SocketWritableByteChannel.java
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/Limits.java
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/netty_unix_util.h

* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/LimitsStaticallyReferencedJniMethods.java
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/PeerCredentials.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright 2020 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* <https://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):

* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/netty_unix.h
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/netty_unix.c
No license file was found, but licenses were detected in source scan.

/*

* Copyright 2021 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* <https://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):

* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/DomainDatagramPacket.java

```
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/DomainDatagramChannelConfig.java
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/SegmentedDatagramPacket.java
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/DomainDatagramSocketAddress.java
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/DomainDatagramChannel.java
```

No license file was found, but licenses were detected in source scan.

<!--

~ Copyright 2016 The Netty Project

~

~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:

~

~ <https://www.apache.org/licenses/LICENSE-2.0>

~

~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

-->

Found in path(s):

```
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/META-INF/maven/io.netty/netty-transport-native-unix-common/pom.xml
```

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2022 The Netty Project

*

* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:

*

* <https://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.

*/

Found in path(s):

* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/RawUnixChannelOption.java
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/GenericUnixChannelOption.java
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/IntegerUnixChannelOption.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2015 The Netty Project

*

* The Netty Project licenses this file to you under the Apache License,

* version 2.0 (the "License"); you may not use this file except in compliance

* with the License. You may obtain a copy of the License at:

*

* <https://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

* License for the specific language governing permissions and limitations

* under the License.

*/

Found in path(s):

* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/Errors.java

* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/DomainSocketAddress.java

* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/netty_unix_errors.h

* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/netty_unix_socket.c

* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/DatagramSocketAddress.java

* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/DomainSocketReadMode.java

* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/netty_unix_errors.c

* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/FileDescriptor.java

* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/netty_unix_filedescriptor.c

* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/DomainSocketChannel.java

* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/netty_unix_filedescriptor.h

* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-

1-jar/io/netty/channel/unix/NativeInetAddress.java
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/Socket.java
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/ServerDomainSocketChannel.java
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/netty_unix_socket.h
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/UnixChannel.java
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/DomainSocketChannelConfig.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright 2017 The Netty Project  
*  
* The Netty Project licenses this file to you under the Apache License,  
* version 2.0 (the "License"); you may not use this file except in compliance  
* with the License. You may obtain a copy of the License at:  
*  
* https://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
* License for the specific language governing permissions and limitations  
* under the License.  
*/
```

Found in path(s):

* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/netty_unix_jni.h
* /opt/cola/permits/1498798466_1670284051.4512854/0/netty-transport-native-unix-common-4-1-85-final-sources-1-jar/io/netty/channel/unix/UnixChannelUtil.java

1.172 checker-qual 2.5.5

1.172.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (c) 2005 Brian Goetz and Tim Peierls  
* Released under the Creative Commons Attribution License  
* (http://creativecommons.org/licenses/by/2.5)  
* Official home: http://www.jcip.net  
*  
* Any republication or derived work distributed in source code form
```

* must include this copyright and license notice.

*/

Found in path(s):

* /opt/ws_local/PERMITS_SQL/1046386584_1590042070.56/0/checker-qual-2-5-5-sources-jar/org/checkerframework/checker/lock/qual/GuardedBy.java

1.173 gopherjs 20181205-snapshot-0766667c

1.173.1 Available under license :

Copyright (c) 2016 Richard Musiol. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2013 Richard Musiol. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.174 docker-java-transport-zero-dep 3.2.13

1.174.1 Available under license :

Apache Commons Codec
Copyright 2002-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

`src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java`
contains test data from <http://aspell.net/test/orig/batch0.tab>.
Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

=====

The content of package `org.apache.commons.codec.language.bm` has been translated from the original php source code available at <http://stevemorse.org/phoneticinfo.htm> with permission from the original authors.
Original source copyright:
Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

Apache HttpClient
Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and

limitations under the License.

1.175 protobuf-java 3.21.9

1.175.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
// Copyright 2008 Google Inc. All rights reserved.  
// Redistribution and use in source and binary forms, with or without  
// modification, are permitted provided that the following conditions are  
// * Redistributions of source code must retain the above copyright  
// notice, this list of conditions and the following disclaimer.  
// * Redistributions in binary form must reproduce the above  
// copyright notice, this list of conditions and the following disclaimer  
// in the documentation and/or other materials provided with the  
// * Neither the name of Google Inc. nor the names of its  
// this software without specific prior written permission.
```

Found in path(s):

```
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-  
jar/com/google/protobuf/GeneratedMessage.java  
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-  
jar/com/google/protobuf/LazyStringList.java  
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-  
jar/com/google/protobuf/MapEntry.java  
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-  
jar/com/google/protobuf/MutabilityOracle.java  
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-  
jar/com/google/protobuf/ArrayDecoders.java  
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-  
jar/com/google/protobuf/UnknownFieldSet.java  
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-  
jar/com/google/protobuf/Writer.java  
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-  
jar/com/google/protobuf/MapFieldLite.java  
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-  
jar/com/google/protobuf/LazyFieldLite.java  
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-  
jar/com/google/protobuf/ProtobufArrayList.java  
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-  
jar/com/google/protobuf/SchemaUtil.java  
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-  
jar/com/google/protobuf/TextFormatParseInfoTree.java  
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-  
jar/google/protobuf/duration.proto  
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-  
jar/com/google/protobuf/ProtocolStringList.java
```

* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/UnknownFieldSchema.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/MapField.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/CodedOutputStream.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/DescriptorMessageInfoFactory.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/RpcUtil.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/RepeatedFieldBuilderV3.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/CodedInputStream.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/CodedInputStreamReader.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/MessageLiteOrBuilder.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/UnsafeByteOperations.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/SingleFieldBuilder.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/NewInstanceSchemas.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/ExtensionSchema.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/ProtobufLists.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/google/protobuf/source_context.proto
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/BooleanArrayList.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/InlineMe.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/CompileTimeConstant.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/google/protobuf/timestamp.proto
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/Message.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/NioByteString.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/MapFieldSchemaLite.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/CheckReturnValue.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/GeneratedMessageInfoFactory.java

* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/MessageSetSchema.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/NewInstanceSchemaFull.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/AllocatedBuffer.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/ExtensionSchemaLite.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/MessageLiteToString.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/ExtensionRegistryFactory.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/ExperimentalApi.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/MessageInfoFactory.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/RepeatedFieldBuilder.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/RopeByteString.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/google/protobuf/empty.proto
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/MapFieldSchemas.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/ExtensionSchemaFull.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/FloatArrayList.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/google/protobuf/wrappers.proto
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/ExtensionSchemas.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/GeneratedMessageV3.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/StructuralMessageInfo.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/RpcCallback.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/CodedOutputStreamWriter.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/ListFieldSchema.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/AbstractMessageLite.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/UninitializedMessageException.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/FieldSet.java

* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/SchemaFactory.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/AbstractParser.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/Reader.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/CanIgnoreReturnValue.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/IntArrayList.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/google/protobuf/any.proto
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/google/protobuf/compiler/plugin.proto
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/DiscardUnknownFieldsParser.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/ExtensionRegistryLite.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/Schema.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/TypeRegistry.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/NewInstanceSchemaLite.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/ByteOutput.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/RawMessageInfo.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/MessageLite.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/Protobuf.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/BlockingService.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/MessageInfo.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/ProtocolMessageEnum.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/ByteBufferWriter.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/MapEntryLite.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/Descriptors.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/InvalidProtocolBufferException.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/MessageOrBuilder.java

* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/google/protobuf/struct.proto
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/Service.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/google/protobuf/type.proto
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/DoubleArrayList.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/ProtoSyntax.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/LongArrayList.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/RpcChannel.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/GeneratedMessageLite.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/UnknownFieldSetLite.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/DynamicMessage.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/google/protobuf/api.proto
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/UnsafeUtil.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/TextFormatParseLocation.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/BlockingRpcChannel.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/Utf8.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/Extension.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/IterableByteBufferInputStream.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/ExtensionLite.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/MapFieldSchemaFull.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/NewInstanceSchema.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/Internal.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/LazyField.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/SingleFieldBuilderV3.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/JavaType.java

* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/LazyStringArrayList.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/FieldType.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/RpcController.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/TextFormat.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/Android.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/SmallSortedMap.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/UnknownFieldSetSchema.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/google/protobuf/field_mask.proto
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/Parser.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/BinaryWriter.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/WireFormat.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/MapFieldSchema.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/MessageSchema.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/PrimitiveNonBoxingCollection.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/AbstractMessage.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/FieldInfo.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/ManifestSchemaFactory.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/OneofInfo.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/BinaryReader.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/ServiceException.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/google/protobuf/descriptor.proto
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/MessageReflection.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/UnknownFieldSetLiteSchema.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/TextFormatEscaper.java

* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/ExtensionRegistry.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/AbstractProtobufList.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/ByteString.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/UnmodifiableLazyStringList.java
* /opt/cola/permits/1473598154_1668495239.2887995/0/protobuf-java-3-21-9-sources-1-jar/com/google/protobuf/BufferAllocator.java

1.176 mockito v3.11.2

1.176.1 Available under license :

The MIT License

Copyright (c) 2007 Mockito contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.177 patrickmn-go-cache 2.1.0

1.177.1 Available under license :

This is a list of people who have contributed code to go-cache. They, or their employers, are the copyright holders of the contributed code. Contributed code is subject to the license restrictions listed in LICENSE (as they were when the code was contributed.)

Dustin Sallings <dustin@spy.net>

Jason Mooberry <jasonmoo@me.com>

Sergey Shepelev <temotor@gmail.com>

Alex Edwards <ajmedwards@gmail.com>

Copyright (c) 2012-2017 Patrick Mylund Nielsen and the go-cache contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.178 aws-java-sdk-::-third-party-::-jackson-core 2.17.122

1.178.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007.

It is currently developed by a community of developers.

Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0
To find the details that apply to this artifact see the accompanying LICENSE file.

Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by
Amazon Technologies, Inc (<http://www.amazon.com/>).

THIRD PARTY COMPONENTS

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:
Apache Commons Lang
Copyright 2001-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.179 aws-java-sdk-::arns 2.17.122

1.179.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade

names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

Note: Other license terms may apply to certain, identified software files contained within or distributed

with the accompanying software if such terms are included in the directory containing the accompanying software.

Such other license terms will then apply in lieu of the terms of the software license above.

AWS SDK for Java 2.0

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by

Amazon Technologies, Inc (<http://www.amazon.com/>).

THIRD PARTY COMPONENTS

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.
- Apache Commons Lang - <https://github.com/apache/commons-lang>
- Netty Reactive Streams - <https://github.com/playframework/netty-reactive-streams>
- Jackson-core - <https://github.com/FasterXML/jackson-core>
- Jackson-dataformat-cbor - <https://github.com/FasterXML/jackson-dataformats-binary>

The licenses for these third party components are included in LICENSE.txt

- For Apache Commons Lang see also this required NOTICE:

Apache Commons Lang

Copyright 2001-2020 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (<https://www.apache.org/>).

1.180 io-grpc-grpc-api 1.39.0

1.180.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2016-2020 Istio Authors

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Envoy

Copyright 2016-2019 Envoy Project Authors

Licensed under Apache License 2.0. See LICENSE for terms.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or

for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason

of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner].

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

protoc-gen-validate

Copyright 2019 Envoy Project Authors

Licensed under Apache License 2.0. See LICENSE for terms.

zero-allocation-hashing

Copyright 2015 Higher Frequency Trading <http://www.higherfrequencytrading.com>

Licensed under Apache License 2.0. See LICENSE for terms.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

/*

* Copyright 2015 The gRPC Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Copyright 2014 The gRPC Authors

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This product contains a modified portion of 'OkHttp', an open source
HTTP & SPDY client for Android and Java applications, which can be obtained
at:

* LICENSE:
* [okhttp/third_party/okhttp/LICENSE](https://github.com/square/okhttp) (Apache License 2.0)
* HOMEPAGE:
* <https://github.com/square/okhttp>
* LOCATION_IN_GRP: C:
* [okhttp/third_party/okhttp](https://github.com/square/okhttp)

This product contains a modified portion of 'Envoy', an open source
cloud-native high-performance edge/middle/service proxy, which can be
obtained at:

* LICENSE:
* [xds/third_party/envoy/LICENSE](https://github.com/envoyproxy/envoy) (Apache License 2.0)
* NOTICE:
* [xds/third_party/envoy/NOTICE](https://github.com/envoyproxy/envoy)
* HOMEPAGE:

- * <https://www.envoyproxy.io>
- * LOCATION_IN_GRPC:
- * xds/third_party/envoy

This product contains a modified portion of 'protoc-gen-validate (PGV)', an open source protoc plugin to generate polyglot message validators, which can be obtained at:

- * LICENSE:
- * xds/third_party/protoc-gen-validate/LICENSE (Apache License 2.0)
- * NOTICE:
- * xds/third_party/protoc-gen-validate/NOTICE
- * HOMEPAGE:
- * <https://github.com/envoyproxy/protoc-gen-validate>
- * LOCATION_IN_GRPC:
- * xds/third_party/protoc-gen-validate

This product contains a modified portion of 'udpa', an open source universal data plane API, which can be obtained at:

- * LICENSE:
- * xds/third_party/udpa/LICENSE (Apache License 2.0)
- * HOMEPAGE:
- * <https://github.com/cncf/udpa>
- * LOCATION_IN_GRPC:
- * xds/third_party/udpa

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing

the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.181 mac-os 4.1.85.Final

1.181.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2019 The Netty Project

*

* The Netty Project licenses this file to you under the Apache License,

* version 2.0 (the "License"); you may not use this file except in compliance

* with the License. You may obtain a copy of the License at:
*
* <https://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):

* /opt/cola/permits/1498798568_1670284042.765666/0/netty-resolver-dns-classes-macos-4-1-85-final-sources-1-jar/io/netty/resolver/dns/macos/MacOSDnsServerAddressStreamProvider.java
* /opt/cola/permits/1498798568_1670284042.765666/0/netty-resolver-dns-classes-macos-4-1-85-final-sources-1-jar/io/netty/resolver/dns/macos/package-info.java
* /opt/cola/permits/1498798568_1670284042.765666/0/netty-resolver-dns-classes-macos-4-1-85-final-sources-1-jar/io/netty/resolver/dns/macos/DnsResolver.java

No license file was found, but licenses were detected in source scan.

<!--

~ Copyright 2021 The Netty Project

~

~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:

~

~ <https://www.apache.org/licenses/LICENSE-2.0>

~

~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

-->

Found in path(s):

* /opt/cola/permits/1498798568_1670284042.765666/0/netty-resolver-dns-classes-macos-4-1-85-final-sources-1-jar/META-INF/maven/io.netty/netty-resolver-dns-classes-macos/pom.xml

1.182 yaml-for-go v2.4.0

1.182.1 Available under license :

Copyright 2011-2016 Canonical Ltd.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed

with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The following files were ported to Go from C files of libyaml, and thus are still covered by their original copyright and license:

apic.go
emitterc.go
parserc.go
readerc.go
scannerc.go
writerc.go
yamlh.go
yamlprivateh.go

Copyright (c) 2006 Kirill Simonov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

SOFTWARE.

1.183 protocol-buffer-java-util-package 3.15.6

1.183.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0
Automatic-Module-Name: com.google.protobuf.util
Bnd-LastModified: 1615494724341
Build-Jdk: 1.8.0_181-google-v7
Built-By: acozzette
Bundle-Description: Utilities for Protocol Buffers
Bundle-DocURL: <https://developers.google.com/protocol-buffers/>
Bundle-License: <https://opensource.org/licenses/BSD-3-Clause>
Bundle-ManifestVersion: 2
Bundle-Name: Protocol Buffers [Util]
Bundle-SymbolicName: com.google.protobuf.util
Bundle-Version: 3.15.6
Created-By: Apache Maven Bundle Plugin
Export-Package: com.google.protobuf.util;version="3.15.6";uses:="com.google.protobuf"
Import-Package: com.google.common.base;version="[30.0,31)",com.google.common.io;version="[30.0,31)",com.google.common.math;version="[30.0,31)",com.google.common.primitives;version="[30.0,31)",com.google.gson;version="[2.8,3)",com.google.gson.stream;version="[2.8,3)",com.google.protobuf;version="[3.15,4)"
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.7))"
Tool: Bnd-3.0.0.201509101326

Found in path(s):

* /opt/cola/permits/1411866746_1662690446.511798/0/protobuf-java-util-3-15-6-jar/META-INF/MANIFEST.MF

1.184 cespare-xxhash v2.1.2

1.184.1 Available under license :

Copyright (c) 2016 Caleb Spare

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to

the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.185 google-api-grpc-proto-google-common-protos 2.7.4

1.185.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
// Licensed under the Apache License, Version 2.0 (the "License");  
// you may not use this file except in compliance with the License.  
// You may obtain a copy of the License at  
// http://www.apache.org/licenses/LICENSE-2.0  
// distributed under the License is distributed on an "AS IS" BASIS,
```

Found in path(s):

```
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/client.proto  
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/context.proto  
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/type/date.proto  
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/type/fraction.proto  
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/quota.proto  
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/cloud/audit/audit_log.proto  
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/endpoint.proto  
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/type/interval.proto  
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/monitored_resource.proto  
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/backend.proto
```

* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/longrunning/operations.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/monitoring.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/source_info.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/type/dayofweek.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/metric.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/type/latlng.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/rpc/code.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/rpc/error_details.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/rpc/status.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/type/timeofday.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/type/money.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/routing.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/distribution.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/control.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/auth.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/resource.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/type/postal_address.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/logging/type/http_request.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/type/calendar_period.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/billing.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/annotations.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/type/phone_number.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/logging/type/log_severity.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/documentation.proto

* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/config_change.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/label.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/usage.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/http.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/error_reason.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/rpc/context/attribute_context.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/type/localized_text.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/service.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/system_parameter.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/httpbody.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/type/month.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/log.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/type/decimal.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/logging.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/field_behavior.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/consumer.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/geo/type/viewport.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/cloud/extended_operations.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/type/color.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/type/datetime.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/type/expr.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/api/visibility.proto
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-jar/google/type/quaternion.proto

No license file was found, but licenses were detected in source scan.

```
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
// http://www.apache.org/licenses/LICENSE-2.0
// distributed under the License is distributed on an "AS IS" BASIS,
// agreement (which includes confidentiality provisions). These features may
```

Found in path(s):

```
* /opt/cola/permits/1526005784_1673051611.2559083/0/proto-google-common-protos-2-7-4-
jar/google/api/launch_stage.proto
```

1.186 jackson-databind 2.14.0

1.186.1 Available under license :

Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers.

Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0. To find the details that apply to this artifact see the accompanying LICENSE file.

Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.187 apache-commons-lang 3.12.0

1.187.1 Available under license :

Apache Commons Lang
Copyright 2001-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the

same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.188 google-android-annotations-library

4.1.1.4

1.188.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2012 The Android Open Source Project  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

Found in path(s):

```
* /opt/ws_local/PERMITS_SQL/1014842714_1591897072.99/0/annotations-4-1-1-4-sources-  
jar/android/annotation/SuppressLint.java  
* /opt/ws_local/PERMITS_SQL/1014842714_1591897072.99/0/annotations-4-1-1-4-sources-  
jar/android/annotation/TargetApi.java
```

1.189 asm-tree 9.1

1.189.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bundle-DocURL: <http://asm.ow2.org>

Bundle-License: BSD-3-Clause;link=<https://asm.ow2.io/LICENSE.txt>

Bundle-ManifestVersion: 2

Bundle-Name: org.objectweb.asm.tree

Bundle-RequiredExecutionEnvironment: J2SE-1.5

Bundle-SymbolicName: org.objectweb.asm.tree

Bundle-Version: 9.1.0

Export-Package: org.objectweb.asm.tree;version="9.1";uses:="org.objectweb.asm"

Implementation-Title: Tree API of ASM, a very small and fast Java byte code manipulation framework

Implementation-Version: 9.1

Import-Package: org.objectweb.asm;version="[9.1,10)"

Module-Requires: org.objectweb.asm;transitive=true

Found in path(s):

* /opt/cola/permits/1183890441_1627493647.83/0/asm-tree-9-1-jar/META-INF/MANIFEST.MF

1.190 kaml 0.49.0

1.190.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.191 junit-jupiter-aggregator 5.8.2

1.191.1 Available under license :

Eclipse Public License - v 2.0

=====

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM

CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. Definitions

Contribution means:

* **a)*** in the case of the initial Contributor, the initial content Distributed under this Agreement, and

* **b)*** in the case of each subsequent Contributor:

* **i)*** changes to the Program, and

* **ii)*** additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of **a)*** distributing or **b)*** making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. Grant of Rights

* **a)*** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

****b)**** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

****c)**** Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

****d)**** Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

****e)**** Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. Requirements

****3.1)**** If a Contributor Distributes the Program in any form, then:

****a)**** the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

****b)**** the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

****i)**** effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

****ii)**** effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

****iii)**** does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

****iv)**** requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

****3.2)**** When the Program is Distributed as Source Code:

****a)**** it must be made available under this Agreement, or if the Program ****i)**** is combined with other material in a separate file or files made available under a Secondary License, and ****ii)**** the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available

under the terms of such Secondary Licenses, and

* **b)*** a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: **a)** promptly notify the Commercial Contributor in writing of such claim, and **b)** allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR

OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

> This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Open Source Licenses

=====

This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the subcomponent's license, as noted in the LICENSE-<subcomponent>.md files.

1.192 commons-codec 1.15

1.192.1 Available under license :

Apache Commons Codec

Copyright 2002-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java
contains test data from <http://aspell.net/test/orig/batch0.tab>.
Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

=====

The content of package org.apache.commons.codec.language.bm has been translated from the original php source code available at <http://stevemorse.org/phoneticinfo.htm> with permission from the original authors.

Original source copyright:

Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. **Grant of Copyright License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. **Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.193 kqueue 4.1.85.Final

1.193.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Implementation-Title: Netty/Transport/Native/KQueue

Bundle-Description: Netty is an asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers and clients.

Automatic-Module-Name: io.netty.transport.kqueue.osx.aarch_64

Bundle-License: <https://www.apache.org/licenses/LICENSE-2.0>

Bundle-SymbolicName: io.netty.transport-native-kqueue

Implementation-Version: 4.1.85.Final

Built-By: chris

Bnd-LastModified: 1668019137616

Bundle-ManifestVersion: 2

Implementation-Vendor-Id: io.netty

Bundle-DocURL: <https://netty.io/>

Bundle-Vendor: The Netty Project

Import-Package: sun.nio.ch;resolution:=optional,org.eclipse.jetty.npn;version="[1,2)";resolution:=optional,org.eclipse.jetty.alpn;version="[1,2)";resolution:=optional

Tool: Bnd-2.4.1.201501161923

Implementation-Vendor: The Netty Project

Bundle-Name: Netty/Transport/Native/KQueue

Bundle-Version: 4.1.85.Final

Created-By: Apache Maven Bundle Plugin

Build-Jdk: 1.8.0_312

Implementation-URL: <https://netty.io/netty-transport-native-kqueue/>

Found in path(s):

* /opt/cola/permits/1498798526_1670358247.4415317/0/netty-transport-native-kqueue-4-1-85-final-jar/META-INF/MANIFEST.MF

No license file was found, but licenses were detected in source scan.

<!--

~ Copyright 2016 The Netty Project

~

~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:

~

~ <https://www.apache.org/licenses/LICENSE-2.0>

~

~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

-->

Found in path(s):

* /opt/cola/permits/1498798526_1670358247.4415317/0/netty-transport-native-kqueue-4-1-85-final-jar/META-INF/maven/io.netty/netty-transport-native-kqueue/pom.xml

1.194 apache-commons-validator 1.6

1.194.1 Available under license :

Apache Commons Validator

Copyright 2001-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.195 aws-java-sdk-secretsmanager 1.11.409

1.195.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2013-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance
 * with
 * the License. A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR
 * CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing
 * permissions
 * and limitations under the License.
 */
/**
```

* <p>

* Creates a new secret. A secret in Secrets Manager consists of both the protected secret data and the important information needed to manage the secret.

* </p>

* <p>

* Secrets Manager stores the encrypted secret data in one of a collection of "versions" associated with the secret. Each version contains a copy of the encrypted secret data. Each version is associated with one or more "staging labels" that identify where the version is in the rotation cycle. The `SecretVersionsToStages` field of the secret contains the mapping of staging labels to the active versions of the secret. Versions without a staging label are considered deprecated and are not included in the list.

* </p>

* <p>

* You provide the secret data to be encrypted by putting text in either the `SecretString` parameter or

or

* binary data in the `SecretBinary` parameter, but not both. If you include `SecretString` or

* `SecretBinary` then Secrets Manager also creates an initial secret version and automatically attaches

* the staging label `AWSCURRENT` to the new version.

* </p>

* <note>

*

*

* <p>

* If you call an operation that needs to encrypt or decrypt the `SecretString` or `SecretBinary` for a secret in the same account as the calling user and that secret doesn't specify a AWS KMS encryption key, Secrets Manager uses the account's default AWS managed customer master key (CMK) with the

* alias `aws/secretsmanager`. If this key doesn't already exist in your account then Secrets Manager creates it for you automatically. All users in the same AWS account automatically have access to use the default CMK. Note that if an Secrets Manager API call results in AWS having to create the account's AWS-managed CMK, it

* can result in a one-time significant delay in returning the result.

* </p>

*

*

* <p>

* If the secret is in a different AWS account from the credentials calling an API that requires encryption or decryption of the secret value then you must create and use a custom AWS KMS CMK because you can't access the

* default CMK for the account using credentials from a different AWS account. Store the ARN of the CMK in the secret when you create the secret or when you update it by including it in the `KMSKeyId`. If you call

* an API that must encrypt or decrypt `SecretString` or `SecretBinary` using credentials from

* a different account then the AWS KMS key policy must grant cross-account access to that other account's user

or

* role for both the kms:GenerateDataKey and kms:Decrypt operations.

* </p>

*

*

* </note>

* <p>

* </p>

* <p>

* Minimum permissions

* </p>

* <p>

* To run this command, you must have the following permissions:

* </p>

*

*

* <p>

* secretsmanager:CreateSecret

* </p>

*

*

* <p>

* kms:GenerateDataKey - needed only if you use a customer-managed AWS KMS key to encrypt the secret. You do not

do not

* need this permission to use the account's default AWS managed CMK for Secrets Manager.

* </p>

*

*

* <p>

* kms:Decrypt - needed only if you use a customer-managed AWS KMS key to encrypt the secret. You do not

need this

* permission to use the account's default AWS managed CMK for Secrets Manager.

* </p>

*

*

* <p>

* Related operations

* </p>

*

*

* <p>

* To delete a secret, use <a>DeleteSecret.

* </p>

*

*

* <p>

* To modify an existing secret, use <a>UpdateSecret.

* </p>

*

```

* <li>
* <p>
* To create a new version of a secret, use <a>PutSecretValue</a>.
* </p>
* </li>
* <li>
* <p>
* To retrieve the encrypted secure string and secure binary values, use <a>GetSecretValue</a>.
* </p>
* </li>
* <li>
* <p>
* To retrieve all other details for a secret, use <a>DescribeSecret</a>. This does not include the encrypted secure
* string and secure binary values.
* </p>
* </li>
* <li>
* <p>
* To retrieve the list of secret versions associated with the current secret, use <a>DescribeSecret</a> and examine
* the <code>SecretVersionsToStages</code> response value.
* </p>
* </li>
* </ul>
*
* @param createSecretRequest
* @return A Java Future containing the result of the CreateSecret operation returned by the service.
* @sample AWSSecretsManagerAsync.CreateSecret
* @see <a href="http://docs.aws.amazon.com/goto/WebAPI/secretsmanager-2017-10-17/CreateSecret"
target="_top">AWS
*   API Documentation</a>
*/
/**
* <p>
* Stores a new encrypted secret value in the specified secret. To do this, the operation creates a new version and
* attaches it to the secret. The version can contain a new <code>SecretString</code> value or a new
* <code>SecretBinary</code> value. You can also specify the staging labels that are initially attached to the new
* version.
* </p>
* <note>
* <p>
* The Secrets Manager console uses only the <code>SecretString</code> field. To add binary data to a secret
with
* the <code>SecretBinary</code> field you must use the AWS CLI or one of the AWS SDKs.
* </p>
* </note>
* <ul>
* <li>
* <p>

```

- * If this operation creates the first version for the secret then Secrets Manager automatically attaches the staging label `AWSCURRENT` to the new version.
-
-
-
-
-
- * If another version of this secret already exists, then this operation does not automatically move any staging labels other than those that you explicitly specify in the `VersionStages` parameter.
-
-
-
-
-
- * If this operation moves the staging label `AWSCURRENT` from another version to this version (because you included it in the `StagingLabels` parameter) then Secrets Manager also automatically moves the staging label `AWSPREVIOUS` to the version that `AWSCURRENT` was removed from.
-
-
-
-
-
- * This operation is idempotent. If a version with a `VersionId` with the same value as the `ClientRequestToken` parameter already exists and you specify the same secret data, the operation succeeds but does nothing. However, if the secret data is different, then the operation fails because you cannot modify an existing version; you can only create new ones.
-
-
-
-
-
-
-
-
-
-
- * If you call an operation that needs to encrypt or decrypt the `SecretString` or `SecretBinary` for a secret in the same account as the calling user and that secret doesn't specify a AWS KMS encryption key, Secrets Manager uses the account's default AWS managed customer master key (CMK) with the alias `aws/secretsmanager`. If this key doesn't already exist in your account then Secrets Manager creates it for you automatically. All users in the same AWS account automatically have access to use the default CMK. Note that if an Secrets Manager API call results in AWS having to create the account's AWS-managed CMK, it can result in a one-time significant delay in returning the result.
-
-
-
-
-
-
-
- * If the secret is in a different AWS account from the credentials calling an API that requires encryption or


```

* <li>
* <p>
* To get the details for a secret, use <a>DescribeSecret</a>.
* </p>
* </li>
* <li>
* <p>
* To list the versions attached to a secret, use <a>ListSecretVersionIds</a>.
* </p>
* </li>
* </ul>
*
* @param putSecretValueRequest
* @return A Java Future containing the result of the PutSecretValue operation returned by the service.
* @sample AWSSecretsManagerAsync.PutSecretValue
* @see <a href="http://docs.aws.amazon.com/goto/WebAPI/secretsmanager-2017-10-17/PutSecretValue"
target="_top">AWS
*   API Documentation</a>
*/
/**
* <p>
* Modifies many of the details of the specified secret. If you include a <code>ClientRequestToken</code> and
* <i>either</i> <code>SecretString</code> or <code>SecretBinary</code> then it also creates a new version
attached
* to the secret.
* </p>
* <p>
* To modify the rotation configuration of a secret, use <a>RotateSecret</a> instead.
* </p>
* <note>
* <p>
* The Secrets Manager console uses only the <code>SecretString</code> parameter and therefore limits you to
* encrypting and storing only a text string. To encrypt and store binary data as part of the version of a secret,
* you must use either the AWS CLI or one of the AWS SDKs.
* </p>
* </note>
* <ul>
* <li>
* <p>
* If a version with a <code>VersionId</code> with the same value as the <code>ClientRequestToken</code>
parameter
* already exists, the operation results in an error. You cannot modify an existing version, you can only create a
* new version.
* </p>
* </li>
* <li>
* <p>
* If you include <code>SecretString</code> or <code>SecretBinary</code> to create a new secret version,

```

Secrets

- * Manager automatically attaches the staging label `AWSCURRENT` to the new version.
- *
- *
- *
- * **Note**
- *
- *
- *
- *
- * If you call an operation that needs to encrypt or decrypt the `SecretString` or `SecretBinary` for a secret in the same account as the calling user and that secret doesn't specify a AWS KMS encryption key, Secrets Manager uses the account's default AWS managed customer master key (CMK) with the
 - * alias `aws/secretsmanager`. If this key doesn't already exist in your account then Secrets Manager creates it for you automatically. All users in the same AWS account automatically have access to use the default CMK. Note that if an Secrets Manager API call results in AWS having to create the account's AWS-managed CMK, it
 - * can result in a one-time significant delay in returning the result.
 - *
 - *
 - *
 - *
 - * If the secret is in a different AWS account from the credentials calling an API that requires encryption or decryption of the secret value then you must create and use a custom AWS KMS CMK because you can't access the default CMK for the account using credentials from a different AWS account. Store the ARN of the CMK in the secret when you create the secret or when you update it by including it in the `KMSKeyId`. If you call an API that must encrypt or decrypt `SecretString` or `SecretBinary` using credentials from a different account then the AWS KMS key policy must grant cross-account access to that other account's user role for both the `kms:GenerateDataKey` and `kms:Decrypt` operations.
 - *
 - *
 - *
 - *
 - * **Minimum permissions**
 - *
 - *
 - * To run this command, you must have the following permissions:
 - *
 - *
 - *
 - * `secretsmanager:UpdateSecret`
 - *

- *
- *
- * <p>
- * kms:GenerateDataKey - needed only if you use a custom AWS KMS key to encrypt the secret. You do not need this
- * permission to use the account's AWS managed CMK for Secrets Manager.
- * </p>
- *
- *
- * <p>
- * kms:Decrypt - needed only if you use a custom AWS KMS key to encrypt the secret. You do not need this permission
- * to use the account's AWS managed CMK for Secrets Manager.
- * </p>
- *
- *
- * <p>
- * Related operations
- * </p>
- *
- *
- * <p>
- * To create a new secret, use <a>CreateSecret.
- * </p>
- *
- *
- * <p>
- * To add only a new version to an existing secret, use <a>PutSecretValue.
- * </p>
- *
- *
- * <p>
- * To get the details for a secret, use <a>DescribeSecret.
- * </p>
- *
- *
- * <p>
- * To list the versions contained in a secret, use <a>ListSecretVersionIds.
- * </p>
- *
- *
- *
 - * @param updateSecretRequest
 - * @return A Java Future containing the result of the UpdateSecret operation returned by the service.
 - * @sample AWSSecretsManagerAsync.UpdateSecret
 - * @see AWS API Documentation

*/

Found in path(s):

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/AWSSecretsManagerAsync.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2013-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.

*

* Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance with

* the License. A copy of the License is located at

*

* <http://aws.amazon.com/apache2.0>

*

* or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR

* CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions

* and limitations under the License.

*/

/**

* <p>

* Creates a new secret. A secret in Secrets Manager consists of both the protected secret data and the important information needed to manage the secret.

* </p>

* <p>

* Secrets Manager stores the encrypted secret data in one of a collection of "versions" associated with the secret.

* Each version contains a copy of the encrypted secret data. Each version is associated with one or more

* "staging labels" that identify where the version is in the rotation cycle. The

* `SecretVersionsToStages` field of the secret contains the mapping of staging labels to the active

* versions of the secret. Versions without a staging label are considered deprecated and are not included in the

* list.

* </p>

* <p>

* You provide the secret data to be encrypted by putting text in either the `SecretString` parameter or

* binary data in the `SecretBinary` parameter, but not both. If you include

`SecretString` or

* `SecretBinary` then Secrets Manager also creates an initial secret version and automatically

attaches

* the staging label `AWSCURRENT` to the new version.

* </p>

* <note>

*

*

* <p>

- * If you call an operation that needs to encrypt or decrypt the `SecretString` or
- * `SecretBinary` for a secret in the same account as the calling user and that secret doesn't specify a
- * AWS KMS encryption key, Secrets Manager uses the account's default AWS managed customer master key (CMK) with the
 - * alias `aws/secretsmanager`. If this key doesn't already exist in your account then Secrets Manager
 - * creates it for you automatically. All users in the same AWS account automatically have access to use the default
 - * CMK. Note that if an Secrets Manager API call results in AWS having to create the account's AWS-managed CMK, it
 - * can result in a one-time significant delay in returning the result.

- * If the secret is in a different AWS account from the credentials calling an API that requires encryption or
- * decryption of the secret value then you must create and use a custom AWS KMS CMK because you can't access the
 - * default CMK for the account using credentials from a different AWS account. Store the ARN of the CMK in the
 - * secret when you create the secret or when you update it by including it in the `KMSKeyId`. If you call
 - * an API that must encrypt or decrypt `SecretString` or `SecretBinary` using
 - * credentials from
 - * a different account then the AWS KMS key policy must grant cross-account access to that other account's user
 - or
 - * role for both the `kms:GenerateDataKey` and `kms:Decrypt` operations.

Minimum permissions

To run this command, you must have the following permissions:

- * `secretsmanager:CreateSecret`
- * `kms:GenerateDataKey` - needed only if you use a customer-managed AWS KMS key to encrypt the secret. You do not
- * need this permission to use the account's default AWS managed CMK for Secrets Manager.

- *
- *
- * <p>
- * kms:Decrypt - needed only if you use a customer-managed AWS KMS key to encrypt the secret. You do not need this
- * permission to use the account's default AWS managed CMK for Secrets Manager.
- * </p>
- *
- *
- * <p>
- * Related operations
- * </p>
- *
- *
- * <p>
- * To delete a secret, use <a>DeleteSecret.
- * </p>
- *
- *
- * <p>
- * To modify an existing secret, use <a>UpdateSecret.
- * </p>
- *
- *
- * <p>
- * To create a new version of a secret, use <a>PutSecretValue.
- * </p>
- *
- *
- * <p>
- * To retrieve the encrypted secure string and secure binary values, use <a>GetSecretValue.
- * </p>
- *
- *
- * <p>
- * To retrieve all other details for a secret, use <a>DescribeSecret. This does not include the encrypted secure string and secure binary values.
- * </p>
- *
- *
- * <p>
- * To retrieve the list of secret versions associated with the current secret, use <a>DescribeSecret and examine the <code>SecretVersionsToStages</code> response value.
- * </p>
- *
- *
- *
- * @param createSecretRequest

```

* @return Result of the CreateSecret operation returned by the service.
* @throws InvalidParameterException
*     You provided an invalid value for a parameter.
* @throws InvalidRequestException
*     You provided a parameter value that is not valid for the current state of the resource.</p>
*     <p>
*     Possible causes:
*     </p>
*     <ul>
*     <li>
*     <p>
*     You tried to perform the operation on a secret that's currently marked deleted.
*     </p>
*     </li>
*     <li>
*     <p>
*     You tried to enable rotation on a secret that doesn't already have a Lambda function ARN configured and
*     you didn't include such an ARN as a parameter in this call.
*     </p>
*     </li>
* @throws LimitExceededException
*     The request failed because it would exceed one of the Secrets Manager internal limits.
* @throws EncryptionFailureException
*     Secrets Manager can't encrypt the protected secret text using the provided KMS key. Check that the
*     customer master key (CMK) is available, enabled, and not in an invalid state. For more information, see
*     <a href="http://docs.aws.amazon.com/kms/latest/developerguide/key-state.html">How Key State Affects
Use
*     of a Customer Master Key</a>.
* @throws ResourceExistsException
*     A resource with the ID you requested already exists.
* @throws ResourceNotFoundException
*     We can't find the resource that you asked for.
* @throws MalformedPolicyDocumentException
*     The policy document that you provided isn't valid.
* @throws InternalServiceErrorException
*     An error occurred on the server side.
* @throws PreconditionNotMetException
*     The request failed because you did not complete all the prerequisite steps.
* @sample AWSSecretsManager.CreateSecret
* @see <a href="http://docs.aws.amazon.com/goto/WebAPI/secretsmanager-2017-10-17/CreateSecret"
target="_top">AWS
*     API Documentation</a>
*/
/**
* <p>
* Stores a new encrypted secret value in the specified secret. To do this, the operation creates a new version and
* attaches it to the secret. The version can contain a new <code>SecretString</code> value or a new
* <code>SecretBinary</code> value. You can also specify the staging labels that are initially attached to the new

```

* version.

* </p>

* <note>

* <p>

* The Secrets Manager console uses only the <code>SecretString</code> field. To add binary data to a secret with

* the <code>SecretBinary</code> field you must use the AWS CLI or one of the AWS SDKs.

* </p>

* </note>

*

*

* <p>

* If this operation creates the first version for the secret then Secrets Manager automatically attaches the staging label <code>AWSCURRENT</code> to the new version.

* </p>

*

*

* <p>

* If another version of this secret already exists, then this operation does not automatically move any staging labels other than those that you explicitly specify in the <code>VersionStages</code> parameter.

* </p>

*

*

* <p>

* If this operation moves the staging label <code>AWSCURRENT</code> from another version to this version (because

* you included it in the <code>StagingLabels</code> parameter) then Secrets Manager also automatically moves the

* staging label <code>AWSPREVIOUS</code> to the version that <code>AWSCURRENT</code> was removed from.

* </p>

*

*

* <p>

* This operation is idempotent. If a version with a <code>VersionId</code> with the same value as the <code>ClientRequestToken</code> parameter already exists and you specify the same secret data, the operation

* succeeds but does nothing. However, if the secret data is different, then the operation fails because you cannot modify an existing version; you can only create new ones.

* </p>

*

*

* <note>

*

*

* <p>

* If you call an operation that needs to encrypt or decrypt the <code>SecretString</code> or <code>SecretBinary</code> for a secret in the same account as the calling user and that secret doesn't specify a

- * AWS KMS encryption key, Secrets Manager uses the account's default AWS managed customer master key (CMK) with the
 - * alias `aws/secretsmanager`. If this key doesn't already exist in your account then Secrets Manager
 - * creates it for you automatically. All users in the same AWS account automatically have access to use the default
 - * CMK. Note that if an Secrets Manager API call results in AWS having to create the account's AWS-managed CMK, it
 - * can result in a one-time significant delay in returning the result.
- * If the secret is in a different AWS account from the credentials calling an API that requires encryption or
 - * decryption of the secret value then you must create and use a custom AWS KMS CMK because you can't access the
 - * default CMK for the account using credentials from a different AWS account. Store the ARN of the CMK in the
 - * secret when you create the secret or when you update it by including it in the `KMSKeyId`. If you call
 - * an API that must encrypt or decrypt `SecretString` or `SecretBinary` using
 - * credentials from
 - * a different account then the AWS KMS key policy must grant cross-account access to that other account's user
 - * or
 - * role for both the `kms:GenerateDataKey` and `kms:Decrypt` operations.

Minimum permissions

To run this command, you must have the following permissions:

- * `secretsmanager:PutSecretValue`
- * `kms:GenerateDataKey` - needed only if you use a customer-managed AWS KMS key to encrypt the secret. You do not
 - * need this permission to use the account's default AWS managed CMK for Secrets Manager.

Related operations

* </p>

- *
- *
- * <p>
- * To retrieve the encrypted value you store in the version of a secret, use <a>GetSecretValue.
- * </p>
- *
- *
- * <p>
- * To create a secret, use <a>CreateSecret.
- * </p>
- *
- *
- * <p>
- * To get the details for a secret, use <a>DescribeSecret.
- * </p>
- *
- *
- * <p>
- * To list the versions attached to a secret, use <a>ListSecretVersionIds.
- * </p>
- *
- *
- *
- * @param putSecretValueRequest
- * @return Result of the PutSecretValue operation returned by the service.
- * @throws InvalidParameterException
- * You provided an invalid value for a parameter.
- * @throws InvalidRequestException
- * You provided a parameter value that is not valid for the current state of the resource.</p>
- * <p>
- * Possible causes:
- * </p>
- *
- *
- * <p>
- * You tried to perform the operation on a secret that's currently marked deleted.
- * </p>
- *
- *
- * <p>
- * You tried to enable rotation on a secret that doesn't already have a Lambda function ARN configured and you didn't include such an ARN as a parameter in this call.
- * </p>
- *
- * @throws LimitExceededException
- * The request failed because it would exceed one of the Secrets Manager internal limits.
- * @throws EncryptionFailureException


```

* Secrets Manager can't encrypt the protected secret text using the provided KMS key. Check that the
* customer master key (CMK) is available, enabled, and not in an invalid state. For more information, see
* <a href="http://docs.aws.amazon.com/kms/latest/developerguide/key-state.html">How Key State Affects
Use
* of a Customer Master Key</a>.
* @throws ResourceExistsException
* A resource with the ID you requested already exists.
* @throws ResourceNotFoundException
* We can't find the resource that you asked for.
* @throws InternalServiceErrorException
* An error occurred on the server side.
* @sample AWSSecretsManager.PutSecretValue
* @see <a href="http://docs.aws.amazon.com/goto/WebAPI/secretsmanager-2017-10-17/PutSecretValue"
target="_top">AWS
* API Documentation</a>
*/
/**
* <p>
* Modifies many of the details of the specified secret. If you include a <code>ClientRequestToken</code> and
* <i>either</i> <code>SecretString</code> or <code>SecretBinary</code> then it also creates a new version
attached
* to the secret.
* </p>
* <p>
* To modify the rotation configuration of a secret, use <a>RotateSecret</a> instead.
* </p>
* <note>
* <p>
* The Secrets Manager console uses only the <code>SecretString</code> parameter and therefore limits you to
* encrypting and storing only a text string. To encrypt and store binary data as part of the version of a secret,
* you must use either the AWS CLI or one of the AWS SDKs.
* </p>
* </note>
* <ul>
* <li>
* <p>
* If a version with a <code>VersionId</code> with the same value as the <code>ClientRequestToken</code>
parameter
* already exists, the operation results in an error. You cannot modify an existing version, you can only create a
* new version.
* </p>
* </li>
* <li>
* <p>
* If you include <code>SecretString</code> or <code>SecretBinary</code> to create a new secret version,
Secrets
* Manager automatically attaches the staging label <code>AWSCURRENT</code> to the new version.
* </p>

```

- *
- *
- * <note>
- *
- *
- * <p>
- * If you call an operation that needs to encrypt or decrypt the <code>SecretString</code> or
- * <code>SecretBinary</code> for a secret in the same account as the calling user and that secret doesn't specify a
- * AWS KMS encryption key, Secrets Manager uses the account's default AWS managed customer master key
- (CMK) with the
- * alias <code>aws/secretsmanager</code>. If this key doesn't already exist in your account then Secrets Manager
- * creates it for you automatically. All users in the same AWS account automatically have access to use the default
- * CMK. Note that if an Secrets Manager API call results in AWS having to create the account's AWS-managed
- CMK, it
- * can result in a one-time significant delay in returning the result.
- * </p>
- *
- *
- * <p>
- * If the secret is in a different AWS account from the credentials calling an API that requires encryption or
- * decryption of the secret value then you must create and use a custom AWS KMS CMK because you can't access
- the
- * default CMK for the account using credentials from a different AWS account. Store the ARN of the CMK in the
- * secret when you create the secret or when you update it by including it in the <code>KMSKeyId</code>. If you
- call
- * an API that must encrypt or decrypt <code>SecretString</code> or <code>SecretBinary</code> using
- credentials from
- * a different account then the AWS KMS key policy must grant cross-account access to that other account's user
- or
- * role for both the kms:GenerateDataKey and kms:Decrypt operations.
- * </p>
- *
- *
- * </note>
- * <p>
- * Minimum permissions
- * </p>
- * <p>
- * To run this command, you must have the following permissions:
- * </p>
- *
- *
- * <p>
- * secretsmanager:UpdateSecret
- * </p>
- *
- *
- * <p>

* kms:GenerateDataKey - needed only if you use a custom AWS KMS key to encrypt the secret. You do not need this

- * permission to use the account's AWS managed CMK for Secrets Manager.

* </p>

- *

*

* <p>

* kms:Decrypt - needed only if you use a custom AWS KMS key to encrypt the secret. You do not need this permission

- * to use the account's AWS managed CMK for Secrets Manager.

* </p>

- *

*

* <p>

* Related operations

* </p>

- *

- *

* <p>

* To create a new secret, use <a>CreateSecret.

* </p>

- *

- *

* <p>

* To add only a new version to an existing secret, use <a>PutSecretValue.

* </p>

- *

- *

* <p>

* To get the details for a secret, use <a>DescribeSecret.

* </p>

- *

- *

* <p>

* To list the versions contained in a secret, use <a>ListSecretVersionIds.

* </p>

- *

*

*

* @param updateSecretRequest

* @return Result of the UpdateSecret operation returned by the service.

* @throws InvalidParameterException

* You provided an invalid value for a parameter.

* @throws InvalidRequestException

* You provided a parameter value that is not valid for the current state of the resource.</p>

* <p>

* Possible causes:

* </p>

```

* <ul>
* <li>
* <p>
* You tried to perform the operation on a secret that's currently marked deleted.
* </p>
* </li>
* <li>
* <p>
* You tried to enable rotation on a secret that doesn't already have a Lambda function ARN configured and
* you didn't include such an ARN as a parameter in this call.
* </p>
* </li>
* @throws LimitExceededException
* The request failed because it would exceed one of the Secrets Manager internal limits.
* @throws EncryptionFailureException
* Secrets Manager can't encrypt the protected secret text using the provided KMS key. Check that the
* customer master key (CMK) is available, enabled, and not in an invalid state. For more information, see
* <a href="http://docs.aws.amazon.com/kms/latest/developerguide/key-state.html">How Key State Affects
Use
* of a Customer Master Key</a>.
* @throws ResourceExistsException
* A resource with the ID you requested already exists.
* @throws ResourceNotFoundException
* We can't find the resource that you asked for.
* @throws MalformedPolicyDocumentException
* The policy document that you provided isn't valid.
* @throws InternalServiceErrorException
* An error occurred on the server side.
* @throws PreconditionNotMetException
* The request failed because you did not complete all the prerequisite steps.
* @sample AWSSecretsManager.UpdateSecret
* @see <a href="http://docs.aws.amazon.com/goto/WebAPI/secretsmanager-2017-10-17/UpdateSecret"
target="_top">AWS
* API Documentation</a>
*/

```

Found in path(s):

```
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-
jar/com/amazonaws/services/secretsmanager/AWSSecretsManagerClient.java
```

```
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-
jar/com/amazonaws/services/secretsmanager/AWSSecretsManager.java
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright 2013-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance
with
```

* the License. A copy of the License is located at
*
* <http://aws.amazon.com/apache2.0>
*
* or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR
* CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing
permissions
* and limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-
jar/com/amazonaws/services/secretsmanager/model/transform/PutSecretValueRequestMarshaller.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-
jar/com/amazonaws/services/secretsmanager/model/transform/CancelRotateSecretRequestMarshaller.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-
jar/com/amazonaws/services/secretsmanager/model/transform/UpdateSecretRequestProtocolMarshaller.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-
jar/com/amazonaws/services/secretsmanager/model/DescribeSecretRequest.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-
jar/com/amazonaws/services/secretsmanager/model/GetRandomPasswordResult.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-
jar/com/amazonaws/services/secretsmanager/model/transform/CreateSecretRequestMarshaller.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-
jar/com/amazonaws/services/secretsmanager/model/transform/PutResourcePolicyRequestProtocolMarshaller.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-
jar/com/amazonaws/services/secretsmanager/model/transform/RotateSecretResultJsonUnmarshaller.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-
jar/com/amazonaws/services/secretsmanager/model/transform/UpdateSecretRequestMarshaller.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-
jar/com/amazonaws/services/secretsmanager/model/transform/GetRandomPasswordRequestProtocolMarshaller.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-
jar/com/amazonaws/services/secretsmanager/model/ListSecretVersionIdsRequest.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-
jar/com/amazonaws/services/secretsmanager/model/transform/RotationRulesTypeMarshaller.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-
jar/com/amazonaws/services/secretsmanager/model/Tag.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-
jar/com/amazonaws/services/secretsmanager/model/GetResourcePolicyResult.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-
jar/com/amazonaws/services/secretsmanager/model/transform/PutResourcePolicyRequestMarshaller.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-
jar/com/amazonaws/services/secretsmanager/model/transform/GetResourcePolicyRequestMarshaller.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-
jar/com/amazonaws/services/secretsmanager/model/transform/DescribeSecretRequestMarshaller.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-
jar/com/amazonaws/services/secretsmanager/model/CancelRotateSecretRequest.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/GetResourcePolicyRequest.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/GetSecretValueRequestProtocolMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/TagMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/GetSecretValueResultJsonUnmarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/UpdateSecretVersionStageRequestProtocolMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/DeleteResourcePolicyResult.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/DescribeSecretResult.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/GetResourcePolicyRequestProtocolMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/MalformedPolicyDocumentException.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/ResourceExistsException.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/RotateSecretRequest.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/ListSecretsRequestProtocolMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/AWSSecretsManagerClientBuilder.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/DeleteSecretRequestMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/ListSecretVersionIdsRequestProtocolMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/UpdateSecretResult.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/DeleteResourcePolicyRequestProtocolMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/GetSecretValueResult.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/UntagResourceRequestMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/TagResourceResult.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/PutResourcePolicyResultJsonUnmarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/TagResourceRequestProtocolMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/SecretVersionsListEntryMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/ListSecretVersionIdsResultJsonUnmarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/SecretVersionsListEntryJsonUnmarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/ListSecretVersionIdsResult.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/CreateSecretResultJsonUnmarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/InvalidNextTokenException.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/RotationRulesTypeJsonUnmarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/ListSecretsRequest.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/ListSecretsRequestMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/TagResourceRequestMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/UntagResourceResult.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/EncryptionFailureException.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/PutResourcePolicyResult.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/CreateSecretRequestProtocolMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/DeleteResourcePolicyRequestMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/AWSSecretsManagerException.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/ListSecretsResult.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/GetRandomPasswordResultJsonUnmarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/UpdateSecretVersionStageResult.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/LimitExceededException.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/AWSSecretsManagerAsyncClientBuilder.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/GetRandomPasswordRequestMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/InvalidRequestException.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/UpdateSecretVersionStageRequestMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/DeleteResourcePolicyRequest.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/DeleteSecretRequestProtocolMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/TagJsonUnmarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/ListSecretVersionIdsRequestMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/PutResourcePolicyRequest.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/DeleteSecretResultJsonUnmarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/DeleteResourcePolicyResultJsonUnmarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/DeleteSecretRequest.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/SecretListEntryJsonUnmarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/PutSecretValueRequestProtocolMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/DescribeSecretRequestProtocolMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/DeleteSecretResult.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/RestoreSecretRequestProtocolMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/RotateSecretRequestProtocolMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/TagResourceRequest.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/PutSecretValueResultJsonUnmarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/package-info.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/UntagResourceResultJsonUnmarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/ResourceNotFoundException.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/DescribeSecretResultJsonUnmarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/GetRandomPasswordRequest.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/TagResourceResultJsonUnmarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/SecretVersionsListEntry.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/UpdateSecretRequest.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/GetSecretValueRequestMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/AbstractAWSSecretsManager.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/RotateSecretRequestMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/CancelRotateSecretResultJsonUnmarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/InvalidParameterException.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/RestoreSecretRequest.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/PutSecretValueResult.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/ListSecretsResultJsonUnmarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/RestoreSecretResultJsonUnmarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/RotateSecretResult.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/SecretListEntryMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/CreateSecretRequest.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/CancelRotateSecretResult.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/AbstractAWSSecretsManagerAsync.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/UntagResourceRequest.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/CancelRotateSecretRequestProtocolMarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/DecryptionFailureException.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/SecretListEntry.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/PreconditionNotMetException.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/CreateSecretResult.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/AWSSecretsManagerAsyncClient.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/GetResourcePolicyResultJsonUnmarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/UpdateSecretVersionStageRequest.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/UpdateSecretResultJsonUnmarshaller.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/PutSecretValueRequest.java

* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/RestoreSecretRequestMarshaller.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/InternalServiceErrorException.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/UpdateSecretVersionStageResultJsonUnmarshaller.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/RotationRulesType.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/transform/UntagResourceRequestProtocolMarshaller.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/RestoreSecretResult.java
* /opt/cola/permits/1411866792_1662683683.0473826/0/aws-java-sdk-secretsmanager-1-11-409-sources-jar/com/amazonaws/services/secretsmanager/model/GetSecretValueRequest.java

1.196 kotlin-scripting-jvm 1.7.20

1.196.1 Available under license :

Apache-2.0

1.197 google-go-cmp v0.5.6

1.197.1 Available under license :

Copyright (c) 2017 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.198 junit-platform-junit-platform-commons

1.8.2

1.198.1 Available under license :

Eclipse Public License - v 2.0

=====

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. Definitions

Contribution means:

* **a)** in the case of the initial Contributor, the initial content Distributed under this Agreement, and

* **b)** in the case of each subsequent Contributor:

* **i)** changes to the Program, and

* **j)** additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations,

interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of ****a)**** distributing or ****b)**** making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. Grant of Rights

****a)**** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

****b)**** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

****c)**** Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

****d)**** Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

****e)**** Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. Requirements

****3.1)**** If a Contributor Distributes the Program in any form, then:

* ****a)**** the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

* **b)*** the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

* **i)*** effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

* **ii)*** effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

* **iii)*** does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

* **iv)*** requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

* **a)*** it must be made available under this Agreement, or if the Program **(i)** is combined with other material in a separate file or files made available under a Secondary License, and **(ii)** the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

* **b)*** a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: **a)** promptly notify the Commercial Contributor in writing of such claim, and **b)** allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

> This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Open Source Licenses

=====

This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the subcomponent's license, as noted in the LICENSE-<subcomponent>.md files.

1.199 jackson-bom 2.14.0

1.199.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.200 apache-log4j-slf4j-binding 2.17.1

1.200.1 Available under license :

Apache Log4j SLF4J Binding
Copyright 1999-1969 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object

form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a

file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.201 swift-poet 1.0.0

1.201.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed

as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.202 error_prone_annotations 2.4.0

1.202.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2014 The Error Prone Authors.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-jar/com/google/errorprone/annotations/NoAllocation.java
* /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-jar/com/google/errorprone/annotations/concurrent/LockMethod.java
* /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-jar/com/google/errorprone/annotations/concurrent/UnlockMethod.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2016 The Error Prone Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-jar/com/google/errorprone/annotations/MustBeClosed.java
* /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-jar/com/google/errorprone/annotations/FormatString.java
* /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-jar/com/google/errorprone/annotations/CompatibleWith.java
* /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-jar/com/google/errorprone/annotations/FormatMethod.java
* /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-jar/com/google/errorprone/annotations/RestrictedApi.java
* /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-jar/com/google/errorprone/annotations/DoNotMock.java
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2015 The Error Prone Authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
 * /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-
jar/com/google/errorprone/annotations/CompileTimeConstant.java
 * /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-
jar/com/google/errorprone/annotations/RequiredModifiers.java
 * /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-
jar/com/google/errorprone/annotations/CanIgnoreReturnValue.java
 * /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-
jar/com/google/errorprone/annotations/Immutable.java
 * /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-
jar/com/google/errorprone/annotations/SuppressPackageLocation.java
 * /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-
jar/com/google/errorprone/annotations/ForOverride.java
 * /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-
jar/com/google/errorprone/annotations/Var.java
 * /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-
jar/com/google/errorprone/annotations/concurrent/LazyInit.java
 * /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-
jar/com/google/errorprone/annotations/IncompatibleModifiers.java
No license file was found, but licenses were detected in source scan.
```

```
/*
 * Copyright 2017 The Error Prone Authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
```

- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-jar/com/google/errorprone/annotations/DoNotCall.java
- * /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-jar/com/google/errorprone/annotations/OverridingMethodsMustInvokeSuper.java
- * /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-jar/com/google/errorprone/annotations/CheckReturnValue.java
- * /opt/cola/permits/1274702949_1645233644.02/0/error-prone-annotations-2-4-0-sources-jar/com/google/errorprone/annotations/concurrent/GuardedBy.java

1.203 jetbrains-kotlin-kotlin-stdlib-jdk7 1.4.10

1.203.1 Available under license :

Note that publicsuffices.gz is compiled from The Public Suffix List:
https://publicsuffix.org/list/public_suffix_list.dat

It is subject to the terms of the Mozilla Public License, v. 2.0:
<https://mozilla.org/MPL/2.0/>

1.204 godoc-text v0.1.0

1.204.1 Available under license :

Copyright 2012 Keith Rarick

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.205 io-grpc-grpc-protobuf 1.39.0

1.205.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2017 The gRPC Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1183891218_1655096184.4932175/0/grpc-protobuf-1-39-0-sources-1-jar/io/grpc/protobuf/package-info.java

* /opt/cola/permits/1183891218_1655096184.4932175/0/grpc-protobuf-1-39-0-sources-1-jar/io/grpc/protobuf/StatusProto.java

* /opt/cola/permits/1183891218_1655096184.4932175/0/grpc-protobuf-1-39-0-sources-1-jar/io/grpc/protobuf/ProtoMethodDescriptorSupplier.java

* /opt/cola/permits/1183891218_1655096184.4932175/0/grpc-protobuf-1-39-0-sources-1-jar/io/grpc/protobuf/ProtoServiceDescriptorSupplier.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2014 The gRPC Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1183891218_1655096184.4932175/0/grpc-protobuf-1-39-0-sources-1-jar/io/grpc/protobuf/ProtoUtils.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2016 The gRPC Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1183891218_1655096184.4932175/0/grpc-protobuf-1-39-0-sources-1-jar/io/grpc/protobuf/ProtoFileDescriptorSupplier.java

1.206 jimfs-parent 1.1

1.206.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.207 apiguardian-apiguardian-api 1.1.2

1.207.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.208 go-tomb-tomb 20180513-snapshot-d5d1b582

1.208.1 Available under license :

tomb - support for clean goroutine termination in Go.

Copyright (c) 2010-2011 - Gustavo Niemeyer <gustavo@niemeyer.net>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.209 netty-codec-redis 4.1.85.Final

1.209.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2016 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License, version 2.0 (the
 * "License"); you may not use this file except in compliance with the License. You may obtain a
 * copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-
jar/io/netty/handler/codec/redis/RedisBulkStringAggregator.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-
jar/io/netty/handler/codec/redis/FullBulkStringRedisMessage.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-
jar/io/netty/handler/codec/redis/ArrayHeaderRedisMessage.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-
jar/io/netty/handler/codec/redis/RedisMessage.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-
jar/io/netty/handler/codec/redis/ArrayRedisMessage.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-
jar/io/netty/handler/codec/redis/RedisArrayAggregator.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-
jar/io/netty/handler/codec/redis/SimpleStringRedisMessage.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-
jar/io/netty/handler/codec/redis/package-info.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-
jar/io/netty/handler/codec/redis/FixedRedisMessagePool.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-
jar/io/netty/handler/codec/redis/AbstractStringRedisMessage.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-
jar/io/netty/handler/codec/redis/IntegerRedisMessage.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-
jar/io/netty/handler/codec/redis/RedisConstants.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-
```

jar/io/netty/handler/codec/redis/ErrorRedisMessage.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-jar/io/netty/handler/codec/redis/RedisCodecException.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-jar/io/netty/handler/codec/redis/BulkStringHeaderRedisMessage.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-jar/io/netty/handler/codec/redis/RedisMessagePool.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-jar/io/netty/handler/codec/redis/DefaultBulkStringRedisContent.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-jar/io/netty/handler/codec/redis/LastBulkStringRedisContent.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-jar/io/netty/handler/codec/redis/RedisMessageType.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-jar/io/netty/handler/codec/redis/RedisDecoder.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-jar/io/netty/handler/codec/redis/DefaultLastBulkStringRedisContent.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-jar/io/netty/handler/codec/redis/RedisEncoder.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-jar/io/netty/handler/codec/redis/BulkStringRedisContent.java
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-jar/io/netty/handler/codec/redis/RedisCodecUtil.java
No license file was found, but licenses were detected in source scan.

```
<!--  
~ Copyright 2016 The Netty Project  
~  
~ The Netty Project licenses this file to you under the Apache License,  
~ version 2.0 (the "License"); you may not use this file except in compliance  
~ with the License. You may obtain a copy of the License at:  
~  
~ https://www.apache.org/licenses/LICENSE-2.0  
~  
~ Unless required by applicable law or agreed to in writing, software  
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
~ License for the specific language governing permissions and limitations  
~ under the License.  
-->
```

Found in path(s):
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-jar/META-INF/maven/io.netty/netty-codec-redis/pom.xml
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright 2018 The Netty Project
```

```
*
* The Netty Project licenses this file to you under the Apache License, version 2.0 (the
* "License"); you may not use this file except in compliance with the License. You may obtain a
* copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1498798508_1670284060.0096362/0/netty-codec-redis-4-1-85-final-sources-1-
jar/io/netty/handler/codec/redis/InlineCommandRedisMessage.java
```

1.210 okio 2.8.0

1.210.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2015 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/SegmentedByteString.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-
jar/commonMain/okio/SegmentedByteString.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/ForwardingTimeout.kt
No license file was found, but licenses were detected in source scan.
```

```
/*
```

* Copyright (C) 2019 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/internal/RealBufferedSource.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/internal/RealBufferedSink.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/internal/SegmentedByteString.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/Utf8.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2019 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/internal/Buffer.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/Timeout.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/BufferedSource.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/Sink.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/Buffer.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/RealBufferedSink.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/BufferedSink.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/Source.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/RealBufferedSource.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/Okio.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2014 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/SegmentPool.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/InflaterSource.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/ForwardingSource.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/BufferedSink.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/JvmOkio.kt

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/Segment.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/BufferedSource.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/GzipSource.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/DeflaterSink.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/GzipSink.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/Buffer.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/Source.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/Sink.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/Timeout.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/RealBufferedSource.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/SegmentPool.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/RealBufferedSink.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/AsyncTimeout.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/ForwardingSink.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2018 Square, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/-Util.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/internal/ByteString.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/-Platform.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/-Platform.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/internal/-Utf8.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/ByteString.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2014 Square Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/ByteString.kt
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2018 Square, Inc.

/*

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at

/*

* <http://www.apache.org/licenses/LICENSE-2.0>

/*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/-DeprecatedUpgrade.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/-DeprecatedUtf8.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/PeekSource.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/-DeprecatedOkio.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/Throttler.kt
No license file was found, but licenses were detected in source scan.

/*

* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements. See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at

/*

* <http://www.apache.org/licenses/LICENSE-2.0>

/*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/-Base64.kt
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2016 Square, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/HashingSink.kt

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/Pipe.kt

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/HashingSource.kt

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/Options.kt

1.211 guava-internalfuturefailureaccess-and-internalfutures 1.0.1

1.211.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2018 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

* /opt/cola/permits/1130987386_1612872111.26/0/failureaccess-1-0-1-sources-
jar/com/google/common/util/concurrent/internal/InternalFutureFailureAccess.java
* /opt/cola/permits/1130987386_1612872111.26/0/failureaccess-1-0-1-sources-
jar/com/google/common/util/concurrent/internal/InternalFutures.java

1.212 mockito-inline 3.11.2

1.212.1 Available under license :

The MIT License

Copyright (c) 2007 Mockito contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.213 go-spew v1.1.1

1.213.1 Available under license :

ISC License

Copyright (c) 2012-2016 Dave Collins <dave@davec.name>

Permission to use, copy, modify, and/or distribute this software for any

purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.214 checker-qual 3.12.0

1.214.1 Available under license :

Checker Framework qualifiers

Copyright 2004-present by the Checker Framework developers

MIT License:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.215 io-grpc-grpc-testing 1.39.0

1.215.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.216 io-grpc-grpc-core 1.39.0

1.216.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2016-2020 Istio Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Envoy

Copyright 2016-2019 Envoy Project Authors

Licensed under Apache License 2.0. See LICENSE for terms.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed

as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner].

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

protoc-gen-validate

Copyright 2019 Envoy Project Authors

Licensed under Apache License 2.0. See LICENSE for terms.

zero-allocation-hashing

Copyright 2015 Higher Frequency Trading <http://www.higherfrequencytrading.com>

Licensed under Apache License 2.0. See LICENSE for terms.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of

the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

/*

* Copyright 2015 The gRPC Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Copyright 2014 The gRPC Authors

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

This product contains a modified portion of 'OkHttp', an open source HTTP & SPDY client for Android and Java applications, which can be obtained at:

* LICENSE:

* okhttp/third_party/okhttp/LICENSE (Apache License 2.0)

* HOMEPAGE:

* <https://github.com/square/okhttp>

* LOCATION_IN_GRPC:

* okhttp/third_party/okhttp

This product contains a modified portion of 'Envoy', an open source cloud-native high-performance edge/middle/service proxy, which can be obtained at:

- * LICENSE:
 - * xds/third_party/envoy/LICENSE (Apache License 2.0)
- * NOTICE:
 - * xds/third_party/envoy/NOTICE
- * HOMEPAGE:
 - * <https://www.envoyproxy.io>
- * LOCATION_IN_GRPC:
 - * xds/third_party/envoy

This product contains a modified portion of 'protoc-gen-validate (PGV)', an open source protoc plugin to generate polyglot message validators, which can be obtained at:

- * LICENSE:
 - * xds/third_party/protoc-gen-validate/LICENSE (Apache License 2.0)
- * NOTICE:
 - * xds/third_party/protoc-gen-validate/NOTICE
- * HOMEPAGE:
 - * <https://github.com/envoyproxy/protoc-gen-validate>
- * LOCATION_IN_GRPC:
 - * xds/third_party/protoc-gen-validate

This product contains a modified portion of 'udpa', an open source universal data plane API, which can be obtained at:

- * LICENSE:
 - * xds/third_party/udpa/LICENSE (Apache License 2.0)
- * HOMEPAGE:
 - * <https://github.com/cncf/udpa>
- * LOCATION_IN_GRPC:
 - * xds/third_party/udpa

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.217 cenk/backoff v4.1.0

1.217.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Cenk Alt

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.218 apache-log4j-api 2.19.0

1.218.1 Available under license :

Apache Log4j API

Copyright 1999-2022 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.219 io-grpc-grpc-stub 1.39.0

1.219.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2016-2020 Istio Authors

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Envoy

Copyright 2016-2019 Envoy Project Authors

Licensed under Apache License 2.0. See LICENSE for terms.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner].

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

protoc-gen-validate

Copyright 2019 Envoy Project Authors

Licensed under Apache License 2.0. See LICENSE for terms.

zero-allocation-hashing

Copyright 2015 Higher Frequency Trading <http://www.higherfrequencytrading.com>

Licensed under Apache License 2.0. See LICENSE for terms.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

/*

* Copyright 2015 The gRPC Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Copyright 2014 The gRPC Authors

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This product contains a modified portion of 'OkHttp', an open source
HTTP & SPDY client for Android and Java applications, which can be obtained
at:

* LICENSE:
* okhttp/third_party/okhttp/LICENSE (Apache License 2.0)
* HOMEPAGE:
* <https://github.com/square/okhttp>
* LOCATION_IN_GRPC:
* okhttp/third_party/okhttp

This product contains a modified portion of 'Envoy', an open source
cloud-native high-performance edge/middle/service proxy, which can be
obtained at:

* LICENSE:
* xds/third_party/envoy/LICENSE (Apache License 2.0)
* NOTICE:
* xds/third_party/envoy/NOTICE
* HOMEPAGE:
* <https://www.envoyproxy.io>
* LOCATION_IN_GRPC:
* xds/third_party/envoy

This product contains a modified portion of 'protoc-gen-validate (PGV)', an open source protoc plugin to generate polyglot message validators, which can be obtained at:

- * LICENSE:
 - * xds/third_party/protoc-gen-validate/LICENSE (Apache License 2.0)
- * NOTICE:
 - * xds/third_party/protoc-gen-validate/NOTICE
- * HOMEPAGE:
 - * <https://github.com/envoyproxy/protoc-gen-validate>
- * LOCATION_IN_GRPC:
 - * xds/third_party/protoc-gen-validate

This product contains a modified portion of 'udpa', an open source universal data plane API, which can be obtained at:

- * LICENSE:
 - * xds/third_party/udpa/LICENSE (Apache License 2.0)
- * HOMEPAGE:
 - * <https://github.com/cncf/udpa>
- * LOCATION_IN_GRPC:
 - * xds/third_party/udpa

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed

as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.220 aws-glue-schema-registry-common

1.1.9

1.220.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<!--
```

```
/*
```

- * Copyright 2020 Amazon.com, Inc. or its affiliates.
- * Licensed under the Apache License, Version 2.0 (the
- * "License"); you may not use this file except in compliance
- * with the License. You may obtain a copy of the License at

*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
-->

Found in path(s):

* /opt/cola/permits/1526005795_1673043477.2135375/0/schema-registry-common-1-1-9-jar/META-INF/maven/software.amazon.glue/schema-registry-common/pom.xml

1.221 junit-jupiter-junit-jupiter-engine 5.8.2

1.221.1 Available under license :

Eclipse Public License - v 2.0

=====

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. Definitions

Contribution means:

- * **a)** in the case of the initial Contributor, the initial content Distributed under this Agreement, and
- * **b)** in the case of each subsequent Contributor:
 - * **i)** changes to the Program, and
 - * **ii)** additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of ****a)**** distributing or ****b)**** making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. Grant of Rights

****a)**** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

****b)**** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

****c)**** Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

****d)**** Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

****e)**** Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. Requirements

****3.1**** If a Contributor Distributes the Program in any form, then:

* ****a)**** the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

* ****b)**** the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

* ****i)**** effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

* ****ii)**** effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

* ****iii)**** does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

* ****iv)**** requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

****3.2**** When the Program is Distributed as Source Code:

* ****a)**** it must be made available under this Agreement, or if the Program ****i)**** is combined with other material in a separate file or files made available under a Secondary License, and ****ii)**** the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

* ****b)**** a copy of this Agreement must be included with each copy of the Program.

****3.3**** Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: ****a)**** promptly notify the Commercial Contributor in writing of such claim, and ****b)**** allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That

Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the

Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

> This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Open Source Licenses

=====

This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the subcomponent's license, as noted in the LICENSE-<subcomponent>.md files.

1.222 pgv-java-parent 0.6.1

1.222.1 Available under license :

protoc-gen-validate

Copyright 2019 Envoy Project Authors

Licensed under Apache License 2.0. See LICENSE for terms.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.223 netty-project 4.1.85.Final

1.223.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<!--
~ Copyright 2014 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ https://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.
-->
```

Found in path(s):

```
* /opt/cola/permits/1498798514_1670358170.6985795/0/netty-resolver-4-1-85-final-jar/META-INF/maven/io.netty/netty-resolver/pom.xml
```

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Implementation-Title: Netty/Resolver

Bundle-Description: Netty is an asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers and clients.

Automatic-Module-Name: io.netty.resolver

Bundle-License: <https://www.apache.org/licenses/LICENSE-2.0>

Bundle-SymbolicName: io.netty.resolver

Implementation-Version: 4.1.85.Final

Built-By: chris

Bnd-LastModified: 1668019011228

Bundle-ManifestVersion: 2

Implementation-Vendor-Id: io.netty

Bundle-DocURL: <https://netty.io/>

Bundle-Vendor: The Netty Project

Import-Package: io.netty.util;version="[4.1,5)",io.netty.util.concurrent;version="[4.1,5)",io.netty.util.internal;version="[4.1,5)",io.netty.util.internal.logging;version="[4.1,5)",sun.nio.ch;resolution:=optional,org.eclipse.jetty.npn;version="[1,2)";resolution:=optional,org.eclipse.jetty.alpn;version="[1,2)";resolution:=optional
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.6))"
Tool: Bnd-2.4.1.201501161923
Implementation-Vendor: The Netty Project
Export-Package: io.netty.resolver;uses:="io.netty.util.concurrent";version="4.1.85"
Bundle-Name: Netty/Resolver
Bundle-Version: 4.1.85.Final
Created-By: Apache Maven Bundle Plugin
Build-Jdk: 1.8.0_312
Implementation-URL: <https://netty.io/netty-resolver/>

Found in path(s):

* /opt/cola/permits/1498798514_1670358170.6985795/0/netty-resolver-4-1-85-final-jar/META-INF/MANIFEST.MF

1.224 testcontainers-junit-jupiter-extension

1.17.3

1.224.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.225 lumberjack v2.0.0

1.225.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Nate Finch

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.226 pmezard-go-difflib 1.0.0

1.226.1 Available under license :

Copyright (c) 2013, Patrick Mezard

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

The names of its contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING

NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.227 golang-protobuf v1.5.2

1.227.1 Available under license :

Copyright 2010 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This source code was written by the Go contributors.

The master list of contributors is in the main Go distribution,

visible at <http://tip.golang.org/CONTRIBUTORS>.

1.228 objenesis 3.2

1.228.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.229 io-grpc-grpc-protobuf-lite 1.39.0

1.229.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright 2014 The gRPC Authors  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

Found in path(s):

```
* /opt/cola/permits/1183892729_1655096107.2169387/0/grpc-protobuf-lite-1-39-0-sources-  
jar/io/grpc/protobuf/lite/ProtoLiteUtils.java  
* /opt/cola/permits/1183892729_1655096107.2169387/0/grpc-protobuf-lite-1-39-0-sources-  
jar/io/grpc/protobuf/lite/ProtoInputStream.java
```

No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright 2017 The gRPC Authors  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.
```

*/

Found in path(s):

* /opt/cola/permits/1183892729_1655096107.2169387/0/grpc-protobuf-lite-1-39-0-sources-jar/io/grpc/protobuf-lite/package-info.java

1.230 grpc-go v1.40.0

1.230.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work

(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"

replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.231 yaml-for-go 20200512-snapshot-9f266ea9

1.231.1 Available under license :

This project is covered by two different licenses: MIT and Apache.

MIT License

The following files were ported to Go from C files of libyaml, and thus are still covered by their original MIT license, with the additional copyright starting in 2011 when the project was ported over:

apic.go emitterc.go parserc.go readerc.go scannerc.go
writerc.go yamlh.go yamlprivateh.go

Copyright (c) 2006-2010 Kirill Simonov

Copyright (c) 2006-2011 Kirill Simonov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License

All the remaining project files are covered by the Apache license:

Copyright (c) 2011-2019 Canonical Ltd

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
Copyright 2011-2016 Canonical Ltd.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.232 protoc-gen-validate v0.6.1

1.232.1 Available under license :

protoc-gen-validate

Copyright 2019 Envoy Project Authors

Licensed under Apache License 2.0. See LICENSE for terms.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade

names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

1.233 prometheus-client v1.12.0

1.233.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Prometheus instrumentation library for Go applications
Copyright 2012-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

The following components are included in this product:

perks - a fork of <https://github.com/bmizerany/perks>
<https://github.com/beorn7/perks>
Copyright 2013-2015 Blake Mizerany, Bjrn Rabenstein
See <https://github.com/beorn7/perks/blob/master/README.md> for license details.

Go support for Protocol Buffers - Google's data interchange format
<http://github.com/golang/protobuf/>
Copyright 2010 The Go Authors
See source code for license details.

Support for streaming Protocol Buffer messages for the Go language (golang).

https://github.com/matttproud/golang_protobuf_extensions

Copyright 2013 Matt T. Proud

Licensed under the Apache License, Version 2.0

1.234 matttproud-golang-protobuf-extensions

v1.0.1

1.234.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work

(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}"

replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright 2012 Matt T. Proud (matt.proud@gmail.com)

1.235 mbknor-jackson-jsonschema 1.0.39

1.235.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 NextGenTel

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.236 junit-5-bill-of-materials 5.8.2

1.236.1 Available under license :

```
import java.io.File
import java.net.URI
```

```
data class License(val name: String, val url: URI, val headerFile: File)
```

```
Apache License
```

```
=====
```

```
_Version 2.0, January 2004_
```

```
_&lt;<https://www.apache.org/licenses/>&gt;_
```

```
### Terms and Conditions for use, reproduction, and distribution
```

```
#### 1. Definitions
```

License shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

Licensor shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

Legal Entity shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, control means **(i)** the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or **(ii)** ownership of fifty percent (50%) or more of the outstanding shares, or **(iii)** beneficial ownership of such entity.

You (or Your) shall mean an individual or Legal Entity exercising permissions granted by this License.

Source form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

Object form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

Work shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

Derivative Works shall mean any work, whether in Source or Object form, that

is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

Contribution shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, submitted means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as Not a Contribution.

Contributor shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- * **(a)** You must give any other recipients of the Work or Derivative Works a copy of this License; and
- * **(b)** You must cause any modified files to carry prominent notices stating that You changed the files; and
- * **(c)** You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- * **(d)** If the Work includes a NOTICE text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Eclipse Public License - v 2.0

=====

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. Definitions

Contribution means:

- * **a)** in the case of the initial Contributor, the initial content Distributed under this Agreement, and
- * **b)** in the case of each subsequent Contributor:
 - * **i)** changes to the Program, and
 - * **ii)** additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular

Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of ****a)**** distributing or ****b)**** making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. Grant of Rights

****a)**** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

****b)**** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

****c)**** Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual

property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

****d)**** Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

****e)**** Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. Requirements

****3.1)**** If a Contributor Distributes the Program in any form, then:

****a)**** the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

****b)**** the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

****i)**** effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

****ii)**** effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

****iii)**** does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

****iv)**** requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

****3.2)**** When the Program is Distributed as Source Code:

****a)**** it must be made available under this Agreement, or if the Program ****i)**** is combined with other material in a separate file or files made available under a Secondary License, and ****ii)**** the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

****b)**** a copy of this Agreement must be included with each copy of the Program.

****3.3)**** Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners

and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: ****a)**** promptly notify the Commercial Contributor in writing of such claim, and ****b)**** allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

> This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Apache License

=====

Version 2.0, January 2004

<<<https://www.apache.org/licenses/>>>>

Terms and Conditions for use, reproduction, and distribution

1. Definitions

License shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

Licensor shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

Legal Entity shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, control means **(i)** the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or **(ii)** ownership of fifty percent (50%) or more of the outstanding shares, or **(iii)** beneficial ownership of such entity.

You (or Your) shall mean an individual or Legal Entity exercising permissions granted by this License.

Source form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

Object form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

Work shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

Derivative Works shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

Contribution shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, submitted means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright

owner as Not a Contribution.

Contributor shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- * **(a)** You must give any other recipients of the Work or Derivative Works a copy of this License; and
- * **(b)** You must cause any modified files to carry prominent notices stating that You changed the files; and
- * **(c)** You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- * **(d)** If the Work includes a NOTICE text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the

Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental,

or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets `[]` replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same printed page as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Open Source Licenses

=====

This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the

subcomponent's license, as noted in the LICENSE-<subcomponent>.md files.

[[contributors]]
== Contributors

Browse the {junit5-repo}/graphs/contributors[current list of contributors] directly on GitHub.

1.237 hamcrest 1.3

1.237.1 Available under license :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>Common Public License - v 1.0</TITLE>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</HEAD>

<BODY BGCOLOR="#FFFFFF" VLINK="#800000">

<P ALIGN="CENTER"><B>Common Public License - v 1.0</B>
<P><B></B><FONT SIZE="3"></FONT>
<P><FONT SIZE="3"></FONT><FONT SIZE="2">THE ACCOMPANYING PROGRAM IS PROVIDED
UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE,
REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE
OF THIS AGREEMENT.</FONT>
<P><FONT SIZE="2"></FONT>
<P><FONT SIZE="2"><B>1. DEFINITIONS</B></FONT>
<P><FONT SIZE="2">"Contribution" means:</FONT>

<UL><FONT SIZE="2">a) in the case of the initial Contributor, the initial code and documentation distributed
under this Agreement, and<BR CLEAR="LEFT">
b) in the case of each subsequent Contributor:</FONT></UL>

<UL><FONT SIZE="2">i) changes to the Program, and</FONT></UL>

<UL><FONT SIZE="2">ii) additions to the Program;</FONT></UL>

<UL><FONT SIZE="2">where such changes and/or additions to the Program originate from and are distributed by
that particular Contributor. </FONT><FONT SIZE="2">A Contribution 'originates' from a Contributor if it was
added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. </FONT><FONT
SIZE="2">Contributions do not include additions to the Program which: (i) are separate modules of software
distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of
the Program. </FONT></UL>
```

<P>

<P>"Contributor" means any person or entity that distributes the Program.

<P>

<P>"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

<P>

<P>"Program" means the Contributions distributed in accordance with this Agreement.

<P>

<P>"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

<P>

<P>2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

<P>3. REQUIREMENTS

<P>A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

<P>When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

<P><STRIKE></STRIKE>
<P><STRIKE></STRIKE>Contributors
may not remove or alter any copyright notices contained within the Program.

<P>

<P>Each Contributor must identify itself as the originator of its Contribution, if any, in a
manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

<P>

<P>4. COMMERCIAL DISTRIBUTION

<P>Commercial distributors of software may accept certain responsibilities with respect to end
users, business partners and the like. While this license is intended to facilitate the commercial use of the Program,
the Contributor who includes the Program in a commercial product offering should do so in a manner which does
not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a
commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify
every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses")
arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to
the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the
Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses
relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor
must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial
Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement
negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

<P>

<P>For example, a Contributor might include the Program in a commercial product offering,
Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes
performance claims, or offers warranties related to Product X, those performance claims and warranties are such
Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to
defend claims against the other Contributors related to those performance claims and warranties, and if a court
requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those
damages.

<P>

<P>5. NO
WARRANTY

<P>EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS
PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER
EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF
TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each
Recipient is solely responsible for determining the appropriateness of using and
distributing the Program and assumes all risks associated
with its exercise of rights under this Agreement, including but not limited to the risks
and costs of program errors, compliance with applicable laws, damage to or loss of data, <FONT
SIZE="2">programs or equipment, and unavailability or interruption of operations.

<P>

<P>6. DISCLAIMER OF LIABILITY

<P>EXCEPT AS EXPRESSLY SET FORTH IN THIS
AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES

(INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

</BODY>

</HTML>

BSD License

Copyright (c) 2000-2006, www.hamcrest.org

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2000-2003, jMock.org

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of jMock nor the names of its contributors may be used to endorse

or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
```

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>EasyMock License</title>
<link rel="stylesheet" href="easymock.css" />
</head>
```

```
<body><div class="bodywidth">
```

```
<h2>
EasyMock 2 License (MIT License)
</h2>
```

```
<em>Copyright (c) 2001-2006 <a href="http://www.offis.de">OFFIS</a>, <a
href="http://www.tammofreese.de">Tammo Freese</a>.</em>
```

```
<p>
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:
```

```
</p>
```

```
<p>
```

```
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.
```

```
</p>
```

<p>

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

</p>

</div>

</body>

</html>

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct

or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of

this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following

boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.238 apache-commons-digester 1.8.1

1.238.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing

the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Commons Digester

Copyright 2001-2008 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (<http://www.apache.org/>).

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

©2023 Cisco Systems, Inc. All rights reserved.