

審判について

1. ジャッジシステム

提出プログラムは、ジャッジシステム上でコンパイルされ、秘匿された判定用入力に対して実行され、正誤判定がなされます。

1.1. 入力と出力

- プログラムは、すべての入力を **標準入力** (`stdin`、例：C++では `std::cin`、Javaでは `System.in`、Pythonでは `sys.stdin`) から読み込む必要があります。
- プログラムは、すべての出力を **標準出力** (`stdout`、例：C++では `std::cout`、Javaでは `System.out`、Pythonでは `sys.stdout`) に書き込む必要があります。
- **標準エラー出力** (`stderr`、例：C++では `std::cerr`、Javaでは `System.err`、Pythonでは `sys.stderr`) に書き込まれた出力は、判定時に無視されます。
- ファイルを開いたり、作成したり、書き込んだりすることは許可されておらず、実行時エラー（**RUN-ERROR**）となります。
- プログラムは、ステータスコード0で終了する必要があります。ゼロ以外のステータスコードで終了すると、出力が正しくても実行時エラー（**RUN-ERROR**）となります。

1.2. 実行環境（サンドボックス）

プログラムは サンドボックス環境内で実行されます。この環境によってプログラムの実行を隔離することで、システムへの影響を防ぎ、すべての参加者に対して公平かつ平等な条件を保証しています。サンドボックス環境内では、以下の制限があります。

- **メモリ使用量:** メモリ使用量は **2GB** に制限されます。この制限は、プログラムのコード、静的変数、スタック、ヒープ、および（Java や Python などの）仮想マシンのオーバーヘッドなどを含む、プロセス全体に対して適用されます。この制限を超えると、通常、実行時エラー（**RUN-ERROR**）となります。
- **スタックサイズ:** C/C++ および Java/Kotlin では、スタックサイズに特定の制限は設けていません。使用できるスタックサイズは、全体のメモリ制限に依存します。Python では、ランタイムのデフォルト設定によるスタックサイズの制限があります。プログラム中で `sys.setrecursionlimit()` を用いて変更することができますが、全体のメモリ制限を超えてスタックサイズが大きくなることはできません。
- **プロセス/スレッド:**
 - プログラムは **単一のプロセッサコア** で実行されます。複数のスレッドやプロセスの使用は可能ですが、パフォーマンスの向上にはつながらない可能性が高く、推奨されません。
 - プログラムの実行時のプロセスの総数は **64** に制限されています。これには、プログラムのメインプロセスと任意の子プロセス、およびシステムが実行管理のために内部的に使用するプロセスが含まれます。この制限を超えると、実行時エラー（**RUN-ERROR**）となります。

- **外部コマンド:** 外部コマンドの実行（例：C/C++ での `system()`）は制限されており、期待通りに動作しない可能性が高いです。
- **ファイルシステムアクセス:** 前述の通り、ファイルを開き、作成し、書き込むことは、許可されていません。

1.3. 提出とコンパイル

すべての提出は以下の要件を満たす必要があります。

- **提出ファイル:** 1回の提出には、単一または複数のファイルを含めることができます。各ファイル名は半角英数字で始まり、半角英数字と `+._-` のみを含むことができます。
- **ソースコードサイズ:** 1回の提出におけるすべてのソースコードファイルの合計サイズは **256KB** を超えてはなりません。これより大きな提出は拒否されます。
- **コンパイル時間:** 提出プログラムのコンパイルは **30秒** 以内に正常に終了しなければなりません。これより長くかかるコンパイルは中止され、コンパイルエラー（**COMPILER-ERROR**）となります。
- **コンパイル時メモリ制限:** コンパイル中の合計メモリ使用量は **2GB** に制限されます。これを超えるとコンパイルは中止され、コンパイルエラー（**COMPILER-ERROR**）となります。

1.4. プラットフォームについて

ジャッジシステムは、プログラムを Google Compute Engine、C4 マシンタイプ（`c4-highcpu-2`、`turboMode` は `ALL_CORE_MAX` に設定）上で実行します。Google Compute Engine の詳細については、公式サイト¹を確認してください。

1. <https://cloud.google.com/compute/docs/cpu-platforms>

2. コンパイラとオプション

ジャッジシステムは、以下のコンパイラと実行環境（インタプリタ）を以下のオプションで使用します。 "\$@" は提出されたソースファイルの名前に置き換えられます。 "\$DEST" はバイナリの名前（デフォルトでは ./a.out）で、システムが決定します。

C
Version gcc (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0
Compile gcc -x c -g -O2 -std=gnunull -static -o "\$DEST" "\$@" -lm
Run "\$DEST" < <u>infile</u> > <u>outfile</u>
C++
Version g++ (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0
Compile g++ -x c++ -g -O2 -std=gnu++20 -static -o "\$DEST" "\$@"
Run "\$DEST" < <u>infile</u> > <u>outfile</u>
Java
Version OpenJDK 21.0.7 (build 21.0.7+6-Ubuntu-0ubuntu122.04)
Compile javac -encoding UTF-8 -sourcepath . -d . "\$@"
Run java -Dfile.encoding=UTF-8 -XX:+UseSerialGC -Xss64m -Xms1920m -Xmx1920m MainClass < <u>infile</u> > <u>outfile</u>
Python 3 (PyPy)
Version Python 3.9.18 (PyPy 7.3.15 with GCC 10.2.1 20210130 (Red Hat 10.2.1-11))
Compile pypy3 -m py_compile "\$@"
Run pypy3 "\$@" < <u>infile</u> > <u>outfile</u>
Kotlin
Version 1.7.21 (JRE 21.0.7+6-Ubuntu-0ubuntu122.04)
Compile kotlinc -d . "\$@"
Run kotlin -Dfile.encoding=UTF-8 -J-XX:+UseSerialGC -J-Xss64m -J-Xms1920m -J-Xmx1920m MainClass < <u>infile</u> > <u>outfile</u>

Java および Kotlin の提出では、通常、メインクラスに `Main` という名前を付ける必要はありません。提出時に、Web インターフェースがメインクラスファイルを自動検出しようとします。複数のファイルを提出する際には、検出結果が正しいことを再確認してください。

Python では、コンパイルコマンドは `py_compile` モジュールを使用して構文のみを検証します。 *.pyc ファイルは実際の実行では使用されません。

3. 提出に対する判定

ジャッジシステムは審判団が用意した複数の入力に対して提出されたプログラムを実行します。

システムはまず、各入力に対する判定を決定します。次に、それらの判定に基づいて、提出に対する単一の最終判定を決定します。チームには、この最終判定のみが報告され、各入力に対する判定は報告されません。

3.1. 各入力に対する判定

各入力について、ジャッジシステムは以下のいずれかの判定を決定します。

- **CORRECT:** プログラムは正常に制限時間内に実行を終了し、正答を出力しました。
 - 正答は審判団によって用意されています。
 - 少少の空白文字は出力の比較の際に無視されます。例えば、行末の余計な空白1文字や、余計な空行1行は無視されます。
 - 正答が一意に定まらない問題が出題されることがあります（例：出力に小数が含まれる）。この場合、問題文に出力の比較方法が記載されます。
- **TIMELIMIT (Time Limit Exceeded):** プログラムが問題ごとに規定される制限時間以内に実行を終了せず、中止されました。
- **RUN-ERROR (Runtime Error):** プログラムが実行中に異常終了しました。例えば、次のような原因があります。
 - イリーガルオペレーションの実行（例：ゼロ除算）。
 - 無効なメモリアクセスの試行（例：配列の範囲外アクセス、ヌルポインタ参照）。
 - メモリ制限の超過。
 - `assert()` による実行中止。
 - ゼロ以外のステータスコードでの終了（例：C言語の `main` の最後に `return 0;` を忘れる）。ステータスコードは出力結果比較の前に確認されます。
 - その他の理由によるクラッシュ。
- **WRONG-ANSWER:** プログラムは制限時間内にクラッシュすることなく実行を終了しましたが、出力が正答と異なると判定されました。
- **OUTPUT-LIMIT (Output Limit Exceeded):** プログラムが **8MB** を超える出力を起こしました。この上限には標準エラー出力に書き込まれた出力は含まれません。
- **NO-OUTPUT:** プログラムは制限時間内にクラッシュすることなく実行を終了しましたが、標準出力に何も出力しませんでした。

3.2. 提出に対する最終判定

ジャッジシステムが決定する、提出に対する最終判定は次の通りです。

3.2.1. Accepted (正解)

- **CORRECT:** プログラムがすべての入力に対して **CORRECT** となりました。

3.2.2. Rejected (20分のペナルティあり)

システムがいずれかの入力に対して以下の5つの判定のいずれかを決定した場合、その決定された判定が、提出に対する最終判定となります。問題を最終的に正解できた場合、これらの提出1回につき **20分の時間ペナルティ** が発生します。

- **TIMELIMIT**
- **RUN-ERROR**
- **WRONG-ANSWER**
- **OUTPUT-LIMIT**
- **NO-OUTPUT**

複数の入力に対して異なる判定となった場合（例：1つが **TIMELIMIT** で、もう1つが **WRONG-ANSWER**）、システムは決定された判定のいずれか1つを提出に対する最終判定として決定します。ただし、決定された判定のうちどれが選択されるかは保証されません。

3.2.3. Rejected (ペナルティなし)

以下の2つの判定は、システムがプログラムを実行しなかったことを示します。これらの提出には **ペナルティは課されません**。

- **COMPILER-ERROR:** ジャッジシステムがコンパイルに失敗しました。提出詳細ページからコンパイラのエラーメッセージを確認することができます。
- **TOO-LATE:** ジャッジシステムがコンテスト終了後に提出を受け取りました。なお、コンテスト終了前に受け取った提出は、コンテストの終了時刻にかかわらず **TOO-LATE** ではない判定が決定されます。

4. プログラミング言語に関して

審判団は、3つの異なる言語グループ（Java/Kotlin, C/C++, Python）のうち2つ以上の言語グループで全ての問題に正解しています。

5. Pythonユーザーの方へ

構文エラーのみが **COMPILER-ERROR** として報告されます。`NameError` や `ModuleNotFoundError` などの他のエラーは **RUN-ERROR** となり、20分のペナルティが発生します。

スクリプトをインタプリタディレクティブ（`#!` で始まる行、シェバンとも呼ばれる）で始めることは問題ありませんが、必須ではありません。^{*2}

2. DOMjudge の過去のバージョンでは、シェバンを含むスクリプトを拒否することがありました。

5.1. 利用可能なPythonモジュール

__decimal	_rawffi	email	pyexpat
__exceptions__	_resource_build	encodings	pypy_tools
__future__	_resource_cffi	ensurepip	pypyjit
__pypy__	_scproxy	enum	pyrepl
__abc	_sha1	errno	queue
_aix_support	_sha256	faulthandler	quopri
_ast	_sha3	fcntl	random
_audiooop_build	_sha512	filecmp	re
_audiooop_cffi	_signal	fileinput	readline
_blake2	_sitebuiltins	fnmatch	reprlib
_bootlocale	_socket	formatter	resource
_bootsubprocess	_sqlite3	fractions	rlcompleter
_bz2	_sqlite3_build	ftplib	runpy
_cffi_backend	_sqlite3_cffi	functools	sched
_cffi_ssl	_sre	future_builtins	secrets
_codecs	_ssl	gc	select
_codecs_cn	_ssl_build	genericpath	selectors
_codecs_hk	_string	getopt	shelve
_codecs_iso2022	_strptime	getpass	shlex
_codecs_jp	_struct	gettext	shutil
_codecs_kr	_structseq	glob	signal
_codecs_tw	_sysconfigdata	graphlib	site
_collections	_sysconfigdata_linux_x86_64-linux-gnu	greenlet	smtplib
_collections_abc	_syslog_build	grp	smtplib
_compat_pickle	_syslog_cffi	gzip	sndhdr
_compression	_testcapi	hashlib	socket
_contextvars	_testing	heapq	socketserver
_continuation	_testmultiphase	hmac	sqlite3
_cppyy	_testmultiphase_build	html	sre_compile
_crypt	_thread	http	sre_constants
_csv	_threading_local	identity_dict	sre_parse
_ctypes	_tkinter	idlelib	ssl
_ctypes_test	_vmpref	imaplib	stackless
_ctypes_test_build	_warnings	imghdr	stat
_curses	_weakref	imp	statistics
_curses_build	_weakrefset	importlib	string
_curses_cffi	_winapi	inspect	stringprep
_curses_panel	abc	io	struct
_dbm	aifc	ipaddress	subprocess
_decimal_build	antigravity	itertools	sunau
_ffi	argparse	json	symbol
_gdbm	array	keyword	symtable
_gdbm_build	ast	lib2to3	sys
_gdbm_cffi	asynchat	linecache	sysconfig
_hashlib	asyncio	locale	syslog
_hpy_universal	asyncore	logging	tabnanny
_immutables_map	atexit	lzma	tarfile
_imp	audiooop	macpath	telnetlib
_io	base64	macurl2path	tempfile
_jitlog	bdb	mailbox	termios
_locale	binascii	mailcap	test
_lsprof	binhex	marshal	textwrap
_lzma	bisect	math	this
_lzma_build	builtins	mimetypes	threading
_lzma_cffi	bz2	mmap	time
_markupbase	cProfile	modulefinder	timeit
_marshal	calendar	msilib	tkinter
_md5	cffi	msvcrt	token
_minimal_curses	cgi	multiprocessing	tokenize
_multibytecodec	cgitb	netrc	tputil

_multiprocessing	chunk	nntplib	trace
_opcode	cmath	ntpath	traceback
_operator	cmd	nturl2path	tracemalloc
_osx_support	code	numbers	tty
_overlapped	codecs	opcode	turtle
_pickle_support	codeop	operator	turtledemo
_posixshmem	collections	optparse	types
_posixshmem_build	colorsys	os	typing
_posixshmem_cffi	compileall	parser	unicodedata
_posixsubprocess	concurrent	pathlib	unittest
_pwdgrp_build	configparser	pdb	urllib
_pwdgrp_cffi	contextlib	pickle	uu
_py_abc	contextvars	pickletools	uuid
_pydecimal	copy	pipes	venv
_pyio	copyreg	pkgutil	warnings
_pypy_generic_alias	cpyext	platform	wave
_pypy_interact	crypt	plistlib	weakref
_pypy irc topic	csv	poplib	webbrowser
_pypy_openssl	ctypes	posix	wsgiref
_pypy_testcapi	ctypes_support	posixpath	xdrlib
_pypy_util_build	curses	pprint	xml
_pypy_util_cffi	dataclasses	profile	xmlrpc
_pypy_util_cffi_inner	datetime	pstats	zipapp
_pypy_wait	dbm	pty	zipfile
_pypy_winbase_build	decimal	pwd	zipimport
_pypy_winbase_cffi	difflib	py_compile	zlib
_pypy_winbase_cffi64	dis	pyclbr	zoneinfo
_pypyjson	distutils	pydoc	
_random	doctest	pydoc_data	