審判について

1. ジャッジシステム

提出されたソースコードは、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 **2 GB** 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++ 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。 Java/Kotlin では、スタックサイズは `-Xss` コマンドラインパラメータにより制限されます。 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()` を用いてこの上限を変更することもできますが、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ASCIIの英数字（半角の`A`～`Z`、`a`～`z`、`0`～`9`）で始まり、ASCIIの英数字と `+.\_-` 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 **256 KB** を超えてはなりません。
- **ソースコードエンコーディング:** すべてのソースコードファイルは UTF-8 でエンコードされている必要があります。他のエンコーディング（例: Shift_JIS）のファイルを提出すると、コンパイルエラー（**COMPILER-ERROR**）や予期しない動作を引き起こす可能性があります。
- **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使用量:** コンパイル中の合計メモリ使用量は **2 GB** 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"` はバイナリの名前で、システムが決定します。

C	
Version	gcc 13
Compile	`gcc -x c -g -O2 -std=gnu11 -static -o "$DEST" "$@" -lm`
Run	`"$DEST" < infile > outfile`
C++	
Version	g++ 13
Compile	`g++ -x c++ -g -O2 -std=gnu++20 -static -o "$DEST" "$@"`
Run	`"$DEST" < infile > outfile`
Java	
Version	OpenJDK 21
Compile	`javac -encoding UTF-8 -sourcepath . -d . "$@"`
Run	`java -Dfile.encoding=UTF-8 -XX:+UseSerialGC -Xss64m -Xms1920m -Xmx1920m MainClass < infile > outfile`
Python 3 (PyPy)	
Version	Python 3.11.11 (PyPy 7.3.18)
Compile	`pypy3 -m py_compile "$@"`
Run	`pypy3 "$@" < infile > outfile`
Kotlin	
Version	1.9.24
Compile	`kotlinc -d . "$@"`
Run	`kotlin -Dfile.encoding=UTF-8 -J-XX:+UseSerialGC -J-Xss64m -J-Xms1920m -J-Xmx1920m MainClass < infile > outfile`

Java および Kotlin の提出には、システムがメインクラスを自動的に検出します。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。

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、提出に対して各入力の判定を決定します。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:** プログラムが制限時間内に実行を終了し、その出力が正解であるとジャッジシステムにより判定されました。
  - 判定方法は審判団が用意しています。これには、正答との比較や、正答が一意に定まらない問題のための検証プログラムなどが含まれます。
  - 通常、多少の空白文字は判定の際に無視されます。例えば、行末の余分な空白や、出力末尾の余分な空行は無視されます。
  - 英字の大文字と小文字の違いは無視されます。例えば、 `Yes` と `yes` は同等とみなされます。
  - 正答が一意に定まらない問題（例：出力に小数が含まれるもの）では、その出力の具体的な判定方法が問題文に記載されます。
- **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。
  - 不正な演算の実行（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（2 GB）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、出力が正解ではないと判定されました。
- **OUTPUT-LIMIT (Output Limit Exceeded):** プログラムが **8 MB** を超える出力をおこないました。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、システムはその判定を、提出に対する最終判定として決定します。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、このリストのものを含む異なる判定が決定された場合（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うちリストの中に含まれるもの1つを、提出に対する最終判定として決定します。ただし、これらの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通常通り判定され、**TOO-LATE** の判定にはなりません。

4. プログラミング言語に関して

審判団は、各問題について、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による正解を確認しています。

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           _pypy_winbase_cffi  dis                 pyexpat
__exceptions__      _pypy_winbase_cffi64 distutils           pypy_tools
__future__          _pypyjson           doctest             pypyjit
__hello__           _random             email               pyrepl
__phello__          _rawffi             encodings           queue
__pypy__            _resource_build     ensurepip           quopri
_abc                _resource_cffi      enum                random
_aix_support        _scproxy            errno               re
_ast                _sha1               faulthandler        readline
_audioop_build      _sha256             fcntl               reprlib
_audioop_cffi       _sha3               filecmp             resource
_blake2             _sha512             fileinput           rlcompleter
_bootsubprocess     _signal             fnmatch             runpy
_bz2                _sitebuiltins       fractions           sched
_cffi_backend       _socket             ftplib              secrets
_cffi_ssl           _sqlite3            functools           select
_codecs             _sqlite3_build      future_builtins     selectors
_codecs_cn          _sqlite3_cffi       gc                  shelve
_codecs_hk          _sre                genericpath         shlex
_codecs_iso2022     _ssl                getopt              shutil
_codecs_jp          _ssl_build          getpass             signal
_codecs_kr          _string             gettext             site
_codecs_tw          _strptime           glob                smtpd
_collections        _struct             graphlib            smtplib
_collections_abc    _structseq          greenlet            sndhdr
_colorize           _sysconfigdata      grp                 socket
_compat_pickle      _syslog_build       gzip                socketserver
_compression        _syslog_cffi        hashlib             sqlite3
_contextvars        _testcapi           heapq               sre_compile
_continuation       _testing            hmac                sre_constants
_cppyy              _testmultiphase     html                sre_parse
_crypt              _testmultiphase_build http                ssl
_csv                _thread             identity_dict       stackless
_ctypes             _threading_local    idlelib             stat
_ctypes_test        _tkinter            imaplib             statistics
_ctypes_test_build  _vmprof             imghdr              string
_curses             _warnings           imp                 stringprep
_curses_build       _weakref            importlib           struct
_curses_cffi        _weakrefset         inspect             subprocess
_curses_panel       _winapi             io                  sunau
_dbm                abc                 ipaddress           symtable
_decimal_build      aifc                itertools           sys
_ffi                antigravity         json                sysconfig
_gdbm               argparse            keyword             syslog
_gdbm_build         array               lib2to3             tabnanny
_gdbm_cffi          ast                 linecache           tarfile
_hashlib            asynchat            locale              telnetlib
_hpy_universal      asyncio             logging             tempfile
_immutables_map     asyncore            lzma                termios
_imp                atexit              mailbox             test
_io                 audioop             mailcap             textwrap
_jitlog             base64              marshal             this
_locale             bdb                 math                threading
_lsprof             binascii            mimetypes           time
_lzma               bisect              mmap                timeit
_lzma_build         builtins            modulefinder        tkinter
_lzma_cffi          bz2                 msilib              token
_markupbase         cProfile            msvcrt              tokenize
_marshal            calendar            multiprocessing     tomllib
_md5                cffi                netrc               tputil
```

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