The 2025 ICPC Latin America Championship

Testing Environment and Submission System

1 Information on the Testing Environment

1.1 Environment

The submission correction system will run on the `Ubuntu GNU/Linux 22.04 LTS amd64` distribution, with the following compilers and interpreters configured:

```
C: gcc version 11.4.0 (Ubuntu 11.4.0-1ubuntu1~22.04)
C++20: gcc version 11.4.0 (Ubuntu 11.4.0-1ubuntu1~22.04)
Python: Python 3.10.12
Java: openjdk 17.0.11
Kotlin: kotlin 1.9.10 (JRE 17.0.11+9-Ubuntu-122.04.1)
```

1.2 Memory limits

```
C, C++20, Python: 1GB
Java, Kotlin: 1GB + 100MB stack
```

1.3 Time limits

Before the contest, the judges will have solved all problems in languages from at least two of the three distinct language groups (C/C++, Java/Kotlin, and Python3). Time limits for each problem will be calculated based on the runtime of those solutions.

A link to the document containing time limits for each problem will be available on the `Problems` tab of Boca's web interface.

1.4 Other limits

```
Source file size: 100KB
Output Size Limit: Each problem has an undisclosed limit on the amount of output
that can be generated, generally on the order of megabytes.

     This limit is generously set to accommodate all valid submissions.
     However, since output to stderr also counts toward this limit,
     excessive debug logging may cause runtime errors. If you encounter
     an unexpected runtime error, consider reducing logging to stderr.
```

1.5 Compilation commands

```
C: gcc -x c -g -O2 -std=gnu11 -static -lm
C++20: g++ -x c++ -g -O2 -std=gnu++20 -static
Java: javac
Python: python3 -m py_compile
Kotlin: kotlinc -J-Xms1024m -J-Xmx1024m -J-Xss100m -include-runtime
```

1.6 C/C++20

- Your program must return a zero, executing, as the last command, `return 0` or `exit(0)`.
- It is known that for problems with very large inputs, `iostream` objects can be slow as, by default, they use a buffer synchronized with the stdio library. If you want to use `cin` and `cout`, disabling such synchronization is advised. This can be achieved by calling `std::ios::sync_with_stdio(false);` at the beginning of your main function. Note that, in this case, using `scanf` and `printf` in the same program should be avoided, since, with separate buffers, misbehavior might occur.

1.7 Java

- DO NOT declare a `package` in your Java program; if you declare a package the program will not execute in Boca.
- Notice that the convention for the solution file name must be obeyed, which means that your public class name must be a capital letter (A, B, C, ...).
- Command for running a Java solution: `java -Xms1024m -Xmx1024m -Xss100m {problem_code}`

1.8 Python

- Note that only Python 3 is supported.
- Python programs will be "syntax checked" when submitted; programs which fail the syntax check will receive a "Compilation Error" verdict.
- Please be aware that Python solutions may not be able to meet the time constraints for some problems. While Python is a versatile and user-friendly language, it may not always provide the best runtime performance for certain tasks.

1.9 Kotlin

- DO NOT declare a `package` in your Kotlin program; if you declare a package the program will not execute in Boca.
- Notice that the convention for the solution file name must be obeyed, which means that your source file must be named (A.kt, B.kt, C.kt, ...).
- Command for running a Kotlin solution: `java -Xms1024m -Xmx1024m -Xss100m {problem_code}Kt`

2 Instructions for the Usage of the Boca Submission System

2.1 Submission of Solutions

To submit a solution, you must use the Boca's web interface:

- Open your browser.
- Login as a team (username and password assigned to your team).
- Access the tab `Runs`. Choose the appropriate problem, the language used and upload the source file.

The verdicts you may receive from the judges are:

- 1 - YES
- 2 - NO - Compilation error
- 3 - NO - Runtime error
- 4 - NO - Time limit exceeded
- 5 - NO - Wrong answer
- 6 - NO - Contact staff

Meanings of 1, 2, 3, 4 and 5 are obvious. 6 is used for unforeseeable circumstances. In this case, use the "Clarifications" menu and provide the "run" number for further clarification.

Your program may be run on multiple input files. Note that this means that if your program has more than one error (say, Time Limit Exceeded and Wrong Answer), then you can get either error as verdict.

There is no such thing as "Presentation Error" or "Format Error". If you misspell the word "impossible", for example, and the problem requires that word as output, then your submission will be judged as "Wrong Answer".

Output formatting should follow the sample output in the problem statement, although extra whitespace within reason is acceptable. For example, if you print out thousands of blank spaces within the time and output limit of the problem, that would be judged as a Wrong Answer; however an extra blank at the end of a line or an extra blank line is acceptable.

If your problem outputs more content than the specified `Output size limit`, your submission will receive a Runtime Error verdict. Note that any content written to the standard error stream also counts toward this limit.

Penalties

Each submission to a problem that receives a "`NO`" verdict before the first "`YES`" verdict for the same problem will incur a time penalty of 20 minutes that is added to the total time of the team. There is no penalty time for unsolved problems.

Exceptions for this rule are "`NO - Compilation error`" and "`NO - Contact staff`" that have no time penalty associated with them.

Response Times

Note that the time required for coming up with a verdict varies depending on the problem, the verdict, and the point at which the contest is in when the submission is received. As solutions are judged simultaneously, this means that you might receive the verdict of your runs out of order. Also, in some cases, manual verification is required from the judges. Depending on what needs to be verified and the number of submissions that require manual verification, even delays on the order of minutes are expected.

2.2 Clarifications

All communication with the judges is done through clarification messages.

To request a clarification concerning a problem statement, you must use Boca's web interface:

- Open your browser.
- Login as a team (username and password assigned to your team).
- Access the tab `Clarifications`. Choose the appropriate problem and type your question.

Both clarification replies from the judges and requests sent by you are displayed there. There will be an alert once your clarification is replied (or the judges issue a global clarification).

Note that it's quite uncommon to have clarifications issued and most of the time the answer to the questions received is already on the problem statement. Please read the problem statement and examine the sample test cases carefully before submitting any request for clarifications.

2.3 Score Board

To visualize the score board showing the teams ranking, you must use Boca's web interface:

- Open your browser.
- Login as a team (username and password assigned to your team).
- Access the tab `Score`. You will have access to the local score board.

2.4 Tasks

The tab `Tasks` allows the team to send files for printing, as well as ask for help from a staff member.

- To print a file, just select it from the disk and click on `Send`.
- To ask for help click on the `S.O.S.` button. Note: the help provided by the staff has to be only related to issues with the computers or other physical problem.