Contestant Handbook

This year's the Regional Contest is online, so the special rules apply:

- Every contestant can use a personal computer (up to three computers per team).
- Contestants are allowed to use any information, including web pages, published before the start of the Contest.
- Contestants are NOT allowed to use help from third parties including source code and other materials created after the beginning of the Contest.
- The jury reserve the right to disqualify any team that violated the previous rule. In case of doubt, it's a team's duty to prove the publication date of used code and materials.

Logins and Passwords

Programming contest management system (PCMS) identifies teams by a login and a password. Logins and passwords will appear on your team page at `https://icpc.global/` on the day of the Contest.

To get your login and password:

1. Enter the registration system `https://icpc.global/login`.
2. Go to *Teams* tab and select your team.
3. Go to *Attachments* tab on the team's page.
4. Download a PDF-file having `RC-PASSWORD` prefix.
5. Downloaded file contains your login and password.

Programming Contest Management System

To start a *PCMS2 Web Client*, open the `https://pcms.itmo.ru/icpc` page in either *Google Chrome* or *Mozilla Firefox*.

You will be prompted for the login and the password. Type your login name and password and press the *Login* button.

Use drop-down list in the upper-right corner to choose *PCMS2 Web Client* language.

Contest Information

The *Information* page contains the Contest information and messages sent by the Jury. During the Contest you will receive messages from the Jury with the results of your runs and clarifications. Incoming messages will be listed on this page.

Current Standings

The *Monitor* page contains the standings table. Teams are displayed as the rows of the table, ordered by team's rank. Problems correspond to the table columns. The intersection of team and problem contains information about team's result for that problem. Possible values are:

"`.`" — team has no runs for this problem;
"`+`" — team has solved this problem the first run was successful;
"`+k`" — team has solved this problem after k unsuccessful runs;
"`-k`" — team has k unsuccessful runs for this problem.
"`?k`" — team has k runs for this problem and the final result is unknown.

The time under this value is the time of the first accepted run on this problem, measured in minutes.

The current standings are unofficial. They will be frozen during the last hour of the Contest. The final standings will be revealed during the Closing Ceremony.

Submit

To submit a program for evaluation, you should select the *Submit* page. Choose the problem you have solved from the *Problem* drop-down list and the language of your solution from the *Language* combo

box. Press the *Choose File* button and choose the file that contains your solution, then press the *Submit solution* button. You will see a page, confirming that your solution was successfully sent to the server for evaluation. Your solution will appear in the *Runs* page. The evaluation result will appear on this page when solution is evaluated.

Clarification Requests

To submit a clarification request, you should select the *Questions* page. Choose the problem you have the clarification request for from the *Problem* combo box and type your clarification request in the *Question* field, then press the *Ask a question* button. The list of already asked questions and corresponding answers are displayed on the same page.

Files

The *Files* page will contain the problem statements and this handbook when the Contest starts.

Run Evaluation

Run is a solution to a problem submitted for judging. The size of the source file with the run may not exceed 256KB.

The run should read the input data from the standard input stream and output result to the standard output stream (it corresponds to input from the keyboard and output to the console if executed manually).

The runs should use the output format specified in the problem statement. The input will have the format specified in the problem statement.

The *memory limit* is the maximum amount of memory that a run may utilize on each test. The *time limit* is the maximum execution time per test. The time and memory limits for each problem are specified in the problem statements. The run is not accepted if the program exceeds these limits.

The run is evaluated by executing it on a secret set of tests, common for all contestants. A run is accepted only if it gives correct answers to all tests.

All test cases are numbered from one. The first test cases in the test set are equal to the sample tests from the problem statement. The following tests are ordered with the idea to make easier test cases come before harder ones, although there are no guarantees.

Runs are evaluated on *Intel Core i3-8100, 3.6GHz* computers under *Windows 10, LTSB 1607*. It may take up to several minutes to evaluate the run, due to the large number of tests. You may solve other problems while waiting for the result.

Programming languages

A solution to a problem is a program written in one of the following programming languages:

Compiler	Compilation command	Execution command
GNU C++ 9.2	`g++ -O2 -std=c++17 -Wl,--stack=67108864 <source file>`	`<compiled program>`
Visual C++	`cl /O2 /EHs /TP <source file>`	`<compiled program>`
Java 11.0.4	`javac <source file>`	`java -Xmx512M -Xss64M <class file>`
Kotlin 1.5.10	`kotlinc <source file>`	`java -Xmx512M -Xss64M <class file>`
Python 3.7.4	Not compiled	`python <source file>`
PyPy 3.6-7.2.0	Not compiled	`pypy <source file>`

Evaluation Results

The *Solutions* page contains run evaluation results and the compiler error messages.

The possible outcomes are listed in the table on the next page. If the run is rejected, the error type and the test number (when applicable) are indicated.

The possible outcomes are listed in their order of priority. For example, if runtime error has occurred, then output is not checked.

Outcome	Test No	Comment	Possible Reasons
Compilation error	No	An executable file was not created after compilation.	Syntax error in the program. Wrong file extension or language specified.
Security violation	Yes	The program tried to violate the Contest Rules.	Error in the program. Attempt to perform a forbidden action.
Time limit exceeded	Yes	The program exceeds the time limit.	Inefficient solution. Error in the program.
Memory limit exceeded	Yes	The program exceeds the memory limit.	Inefficient solution. Error in the program.
Idleness limit exceeded	Yes	The program does not consume processor time for a long period.	Error in the program. Protocol violation for an interactive problem.
Runtime error	Yes	The program terminates with non-zero exit code or throws an uncaught OS exception.	Runtime error. Uncaught exception. Explicit non-zero exit code.
Wrong answer	Yes	The answer is not correct or the output does not match the format specified in the problem statement.	Algorithm is not correct. Output format is not correct. Missed corner case.
Accepted	No	The run is accepted.	Program is correct.

Runs are not allowed to:

- access the network;
- perform any file or network I/O;
- execute other programs and create new processes;
- create or manipulate any GUI resources (windows, dialog boxes, etc.);
- work with external devices (sound, printer, etc.);
- attack system security;
- do anything else that can stir the evaluation process and the Contest.

In the case of the purposeful rules violation the violating team will be disqualified.

Evaluation process may be stopped several minutes before the end of the Contest. All runs submitted after this moment will be evaluated just after the end of the Contest.

Clarification Requests

A contestant may submit a claim of ambiguity or error in a problem statement by submitting a clarification request. Clarification requests are accepted only in English.

If the Jury agrees that an ambiguity or error exists, a clarification will be issued to all contestants. The Jury encourages contestants to use the sample input and output for resolving (apparent) ambiguities.

Scoring of the Contest

Teams are ranked according to the most problems solved. Teams who solve the same number of problems are ranked first by least total time and, if necessary, by the earliest time of submittal of the last accepted run.

The *total time* is the sum of the penalty time for each problem solved. The *penalty time* for a solved problem is the time elapsed from the beginning of the Contest to the submittal of the first accepted run plus 20 penalty minutes for every successfully compiled, but rejected run for that problem before the first accepted run. There is no penalty time for a problem that is not solved. There is no penalty for compilation errors.

Tips and Tricks

Java

The first class defined in your solution must be declared `public` and must contain method `main`. Otherwise, you will receive *Compilation Error* outcome.

`Scanner` class is slow. You can use `BufferedReader` and `StreamTokenizer` classes instead.

Before using `Scanner`, `PrintWriter` and other classes that read or write floating-point numbers, include the following line in your code: "`Locale.setDefault(Locale.US);`".

C++

In case of large input data in MinGW we recommend to and disable synchronization with the standard C streams via "`ios_base::sync_with_stdio(false);`" and use `std::cin`. Moreover, you may disable further synchronization with `scanf` and `printf`, using "`cin.tie(0); cout.tie(0);`".

In Visual C++ the `scanf` is the fastest option.

Interactive Problems

Finish each output with a new line and flush the standard output stream.

The following code will auto-flush on the new line: "`cout << ... << endl`" in *C++*, "`System.out.println`" in *Java*, "`println`" in *Kotlin*, "`print`" in *Python*.

To explicit flush after the other output procedure, use "`fflush(stdout)`" in *C++*, "`System.out.flush()`" in *Java* and *Kotlin*, "`sys.stdout.flush()`" in *Python*.

Common issues for interactive problems:

- *Wrong Answer* — either an incorrect output or the interactive protocol violation.
- *Idleness Limit Exceeded* — the program waits for the input and there are no data in the standard input. Possible causes:
  - the program waits for the input instead of writing the output or finishing the execution;
  - the program has not output the new line or has not flushed the output stream, thus the Jury's program has not received its output.
- *Runtime Error* — rarely an interaction issue. Usually, it's a simple bug in the program.

We wish you good luck and have fun during the Contest!