Contestant Handbook

Conduct of the Contest

The duration of the Contest is five hours. During the Contest each team is provided with one personal computer and a set of problems. At least eight problems and at most fifteen problems will be proposed.

During the competition, contestants solve the proposed problems. A solution to a problem is a program written in one of the supported programming languages. Contestants may solve different problems using different supported programming languages.

Contestants may bring and use unannotated natural language dictionaries (except electronic ones), blank sheets of paper and instruments for writing **only**. Contestants **may not** bring or use any books (except dictionaries), reference manuals, electronic dictionaries, program listings, any machine-readable information (software or data on any kind of storage), computing devices (handhelds, portable PCs, notebooks, calculators, smart watches), mobile phones, or any other communication devices.

During the competition, contestants may communicate to members of their team, members of the Executive Committee of the Jury, the Technical Committee, and the Support Staff **only**.

Before the beginning of the Contest, all computers will be turned on. During the Contest each team will be provided with an envelope containing the problem statements (3 copies).

Contestants **may not** touch the computer or problem statements before the beginning of the Contest. The Contest will begin after the notification "*THE CONTEST IS STARTED*".

Contestants may use a network printer during the Contest. Support Staff delivers printouts to the teams.

In case of any trouble (system related problems, etc.) contestants should ask the Support Staff for help.

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.

Immediately after submission of any run, the team may continue working on other problems.

Contest software evaluates each run and marks it as *accepted* or *rejected*.

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.

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.

As soon as the run is evaluated, the contest software displays evaluation results. The team is informed whether the run is accepted or not. If the run is rejected, the error type and the test number (when applicable) are indicated.

All tests 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.

The possible outcomes are listed in the following table.

Outcome	Test Number	Comment	Possible Reasons
Compilation error	No	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; • purposeful rules violation (the violating team is disqualified in this case).
Runtime error	Yes	The program terminates with non-zero exit code or throws an uncaught OS exception.	• Runtime error; • uncaught exception; • missing '`return 0`' statement in C++ main function; • '`return (non-zero)`' statement in C++ main function; • '`System.exit(non-zero)`' in Java.
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.	• Input from console; • error in the program.
Wrong answer	Yes	The answer is not correct or the checker cannot check output because it does not match the format specified in the problem statement.	• The algorithm is not correct; • output format is not correct; • no output file.
Accepted	No	Run is accepted.	Program is correct.

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

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.

Runs are evaluated on *Intel Core i3-8100, 3.6GHz* computers under *Windows 10, LTSB 1607*.

Runs are not allowed to:

- access the network;
- perform any file or network I/O;
- execute other programs and create new processes;
- work with subdirectories;
- 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.

Programming languages

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

Language	Compilation command
Java 11	`javac <source file>`
Visual C++ 2019	`cl /O2 /TP /EHsc <source file>`
GNU C++ (MinGW 32-bit) 7.3	`g++ -O2 -Wl,--stack=67108864 -x c++ -std=c++17 <source file>`
Kotlin 1.3	`kotlinc <source file>`
Python 3.7	Not compiled

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 a 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.

Practice Session

During the Practice Session teams become familiar with the Contest environment and software solving sample problems (1 to 3 simple problems).

During the Practice Session teams **may not** store any source code anywhere except working directory. Also, you **may not** attach any devices to the computer or alter its hardware configuration.

The results of the Practice Session are not taken into consideration when determining the Contest standings. However, the Executive Committee of the Jury may disqualify from the Contest any team violating the Contest Rules during the Practice Session.

PCMS Web Client User Guide

To start a *PCMS2 Web Client*, open your browser and go to the `https://pcms.itmo.ru/icpc` page. You will be prompted for login name and password. Type your login name and password and press the *Login* button.

The *Information* page contains 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. You can always read all your previous messages in this list.

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.

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

To submit a program for evaluation, you should select the *Submit* page. Choose the problem you have solved from the *Problem* combo box 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. You may solve other problems while waiting for the result.

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.

Java and Kotlin Tips and Tricks

Your solutions will be executed by the command: "`java -Xmx<memory limit> -Xss64m <class file>`". The stack size of all threads created by your program is set to 64Mb.

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++ Tips and Tricks

In case of large input data in MinGW 32-bit 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 futher synchronization with `scanf` and `printf`, using "`cin.tie(0); cout.tie(0);`".

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