Contestant Handbook

Conduct of the Qualification

- Participants may gather in any convenient place, a team may use one personal computer to write and run the code and arbitrary number devices to read the statements and their code.
- Participants are allowed to use any sources published before the competition, including online ones.
- Participants are NOT allowed to use third-party assistance, including using source code and other materials created after the start of the contest.
- Participants are **NOT allowed to use any systems based on artificial intelligence** to solve problems, including, but not limited to:
  - to generate code or pseudocode;
  - to diagnose or troubleshoot errors;
  - to help understand the problem, create logic, or make decisions;

  In particular, it is prohibited to enter the problem statement, its summary, any excerpt, subproblem, or example into the AI system.
- The jury reserves the right to disqualify a team, in case of violation of the specified points.

Username and password

The testing system identifies the team by the username and password, which will be posted on the team page on the day of the contest on the website `https://icpc.global/`.

To obtain them:

1. Log in to the registration system `https://icpc.global/login`.
2. Go to the *Teams* tab and select your team.
3. On the team page, go to the *Attachments* tab.
4. From the list of files that appears, download the PDF file with the prefix `NWQ-PASSWORD`.
5. The downloaded file contains your username and password.

PCMS Web Client User Guide

To start a *PCMS 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 and the time of the first accepted run on this problem, measured in minutes.

Possible results 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.

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.

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

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.	• Protocol violation in an interactive problem; • 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.
Accepted	No	Run is accepted.	Program is correct.

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*.

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:

Compiler	Compilation command	Run command
GNU C++ 13.2	`g++ -O2 -std=c++20 -Wl,--stack=67108864 <source file>`	`<exec file>`
Visual C++ 2019	`cl /O2 /EHs /TP <source file>`	`<exec file>`
Java 21.0.5	`javac <source file>`	`java -Xmx1G -Xss64M <class file>`
Kotlin 2.0.20	`kotlinc <source file>`	`java -Xmx1G -Xss64M <class file>`
Python 3.11.6	*Not compiled*	`python <source file>`
PyPy 7.3 (Py 3.10)	*Not compiled*	`pypy <source file>`

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 and Russian.

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 5 simple problems).

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.

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 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.

Interactive and Run Twice Problems

Do not decouple input/output streams from standard input/output streams.

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.

Programs that violate interaction protocols may produce unpredictable outcomes. A common example is a program that continues to operate after receiving the final result or after exhausting the allowed interaction steps.

Solutions for run twice problems are subject to the same rules, as the run twice mechanism is implemented in the same way as interactive problems. Ensure that you include new lines in your output, particularly the final one.