


MikeMirzayanov's blog

Codeforces Contest Rules

By **MikeMirzayanov**, 13 years ago, translation, 

This document provides a detailed and formal version of the Codeforces contest rules. If you're simply looking to understand the basics, we recommend starting with the post [Codeforces Contests](#).

Please note that organizers may make minor modifications to the rules for specific rounds; any such changes will be included in the official announcement of the respective round. Unless otherwise stated, the following rules should be considered the default for all contests.

By participating in a Codeforces contest and/or submitting any materials, you automatically agree to abide by these [terms and conditions](#).

General information about a round

- You need to register to participate in a round. The registration opens six hours before the start of the contest and closes 5 minutes before it, unless the information about the round states otherwise.
- The system of conducting contests has two interface languages available: Russian and English. The contestants can switch between these two languages.
- A round continues for 2 hours, unless the information about the round states other duration time.
- Contestants are given 5 problems to solve, unless the information about the round states another number of problems.
- Problem statements are available in two languages: Russian and English
- Statements of all problems become available to read at the moment the round starts.
- Before the contest begins, all registered contestants are randomly split into rooms. Each room contains about 40 contestants.

Questions

- During a contest the contestants can ask questions to the contest jury via the system. The questions may concern mistakes, ambiguous spots and inaccuracies in the statement.
- A question must be asked in one of two languages: Russian or English. The question must be clear and written in a correct language. Otherwise, the jury preserves the right not to consider the question and answer 'no comments'.
- The contest jury can answer the given question in an arbitrary text form.
- The answer `No comments` means that the question either doesn't the requirements listed in point 2 or the answer to the question is in the statement or in the contest rules.
- An answer to a question is only available to the contestant who has asked it apart from the cases when the contest jury decides that the question is important for all contestants. In this case the jury can make the question and the answer to it available to all contestants.

Problem Solution

- A problem solution is a program written in one of the following programming languages (different problems can be solved in different programming languages, the list of languages can be widened):
 - C/C++
 - Delphi/Pascal

→ Pay attention

Before contest
[Codeforces Round \(Div. 2\)](#)
 3 days
[Register now »](#)
 *has extra registration?

→ Nova_Storm07

★

Contribution: 0

Nova_Storm07

- [Settings](#)
- [Blog](#)
- [Teams](#)
- [Submissions](#)
- [Talks](#)
- [Contests](#)

→ Top rated

#	User	Rating
1	tourist	3849
2	jiangly	3699
3	orzdevinwang	3695
4	jqdai0815	3682
5	maroonrk	3581
6	Benq	3546
7	Ormlis	3529
8	ksun48	3524
9	ecnerwala	3513
10	Radewoosh	3480
Countries Cities Organizations		View all →

→ Top contributors

#	User	Contrib.
1	cry	162
2	Dominater069	159
3	Qingyu	158
3	-is-this-fft-	158
5	atcoder_official	155
5	djm03178	155
7	errorgorn	153
7	adamant	153
9	luogu_official	150
10	soullless	143
		View all →

→ Find user

- Java
- C#
- Python
- Ruby
- PHP
- OCaml
- Haskell
- Perl
- Scala
- D
- Go

2. Requirements to the solutions:

- The full program should be contained in a single file.
- If the problem statement doesn't specify the names of input or output, you must read the data from the standard input and write it to the standard output.
- You are forbidden to work with the Net.
- You are forbidden to perform input-output operations except for opening, closing, reading and writing files and standard streams given in the problem statements to perform input-output.
- You are forbidden to run other programs and create processes.
- You are forbidden to modify files or directories' permissions in the file system.
- You are forbidden to work with directories other than the current one.
- You are forbidden to work with the operating system registry.
- You are forbidden to create and use GUI elements (windows, dialogs etc.)
- You are forbidden to work with external devices.
- You are forbidden to perform any other actions that can in any manner destabilize the judging process.

3. After a contestant solves a problem, he submits the solution via the contest system interface and continues working on other problems. After the system receives and judges a solution, it immediately responses by displaying the result of judging the solution.

4. A contestant can see the results of judging of his solutions on the corresponding tab of the contest system. During a contest a contestant can see the results of only his solutions' judging.

Compiling solutions

1. As a contestant submits a solution, he chooses a compiler that the judging system should use as it compiles the program. The judging system uses the following compilers for judging (the jury has the right to add other compilers at its discretion or replace the compiler's versions with the newer ones):

- MinGW GNU C++ 4
- MinGW GNU C 4
- MS VS C++ 2010
- Free Pascal 2
- Delphi 7
- C# Mono 2
- Java 6, 7
- Ruby 1
- Python 2
- PHP 5
- Haskell GHC 6
- Objective Caml 3
- Scala 2
- Perl 5
- MS C# .NET 4

Handle:

Find

→ Recent actions

- R_ana → [Just advice](#)
- ShauryaH4X → [Help with 1861C](#)
- eugenechka.boyko.2_0-0 → [Codeforces Round 1022 Editorial](#)
- removed1 → [Codeforces Beta Round #1 - Tutorial](#)
- sr2005roy → [Invitation to Dream In Code '25 — IIIT Lucknow](#)
- sunkuangzheng → [Codeforces Round 1023 \(Div. 2\)](#)
- AlperenT → [IOI Community Discord Server | 2025](#)
- avnithv → [TJIOI Invitational Open in Informatics \(TJIOI\) 2025](#)
- PieArmy → [EJOI 2025 Teams](#)
- donpaul2718 → [Advice for Indians getting into competitive programming.](#)
- PsInd → [Problem recommendation system](#)
- HCCoder → [How can I learn to implement code without bugs?](#)
- adamant → [OCPC 2025 Summer](#)
- aloneguy → [How to get into DSA and Competitive Programming ?](#)
- M.UmairAhmadMirza → [Meet Pakistan's IOI 2025 Team!](#)
- Otherwordly → [IOI 2025](#)
- DilshodbekX → [APIO 2025 Call For Tasks](#)
- husc24.Serge_ → [Please help me with this problem](#)
- atcoder_official → [AtCoder Beginner Contest 404 Announcement](#)
- DanTheMan. → [Montgomery Blair Informatics Tournament 2025](#)
- SpyrosAliv → [Codeforces Round 1020 \(Div. 3\).](#)
- Xanlar → [IP bans are still remaining?](#)
- akshat.jaipuriar → [ML Based Rating Change Predictor and Problem Recommender](#)
- SecondWave → [What is not optimized in this code ?](#)
- ppavic → [Croatian Olympiad in Informatics 2025](#)

[Detailed →](#)

2. The contest organizers are not responsible for problems caused by the compiler's version in the judging system not matching the compiler's version used by the contestant as he was writing the program. To partially avoid such problems, there is an option to run a contestant's code on the server's side. To run the code, the contestant should submit the program and the input to the server. After a while the judging system will display the result of the program's execution and the output.
3. Solutions are compiled and run under Windows OS. Line breaks are set by a couple of symbols #13#10. A file's or an input's last line should end with a line break (right before the end of the file).
4. If a compilation error occurs, than the contestant gets the judging result as 'Compilation error' and the compilation log.
5. The size of the file with the source code shouldn't exceed 64 kilobytes.
6. The judging system uses command line compilers to compile solutions. The compiler command lines for different compilers are given in the table:

Compiler	compilation (execution) command line
GNU C++ 4	g++.exe -static -fno-optimize-sibling-calls -fno-strict-aliasing -DONLINE_JUDGE -lm -s -x c++ -Wl,--stack=268435456 -O2 -o %name%.exe %1
GNU C 4	gcc.exe -static -fno-optimize-sibling-calls -fno-strict-aliasing -DONLINE_JUDGE -fno-asm -lm -s -Wl,--stack=268435456 -O2 -o %name%.exe %1
GNU C++ 4 (C++0x)	g++.exe -static -fno-optimize-sibling-calls -fno-strict-aliasing -DONLINE_JUDGE -lm -s -x c++ -Wl,--stack=268435456 -O2 -std=c++0x -D__USE_MINGW_ANSI_STDIO=0 -o %name%.exe %1
MS VS C++ 2010	cl /W4 /F268435456 /EHsc /O2 /DONLINE_JUDGE %1
Free Pascal 2	fpc -n -O2 -Xs -Sgic -viwn -dONLINE_JUDGE -Cs67107839 -Mdelphi -XS %1 -o%name%.exe
Delphi 7	dcc32 -Q -\$M1048576,67107839 -DONLINE_JUDGE -cc %1
C# Mono 2	dmcs -define:ONLINE_JUDGE -o+ -out:%name% %1
Java 6,7	javac -cp ".;" %1, java.exe -Djava.security.manager -Djava.security.policy=java.policy -javaagent:invokeagent.jar -Xmx512M -Xss64M -DONLINE_JUDGE=true -Duser.language=en -Duser.region=US -Duser.variant=US -jar %s
Ruby 1	ruby.exe -c %1
Python 2	python -c "compile(open('%1').read(), '%1', 'exec')"
PHP 5	php.exe -l %1
Haskell GHC 6	ghc --make -O -o %name% %1
OCaml 3	ocamlopt nums.cmxa str.cmxa -pp camlp4o -unsafe -o %name%.exe-ocaml %1
Scala 2	Cm. Java
Perl	perl.exe %s
D	dmd -L/STACK:268435456 -version=ONLINE_JUDGE -O %s

Compiler	compilation (execution) command line
Go	go build %1

Judging Solutions

1. During the contest the solutions are judged on a small number of tests called pretests.
2. The result of judging the solution on the pretests are displayed to the contestant right after the judging.
3. Each test is represented by the input for the contestant's program. The limits for the input contained in the test are indicated in the problem statement.
4. All problems indicate the maximum time a program can take to work on one test and the maximum memory that can be consumed.
5. Judging is performed by consecutively running the program on each test.
6. The test is considered passed if running the program terminated with the return code 0, the process fit into the given time and memory limits and besides, the program printed the correct answer to the given test. The table below gives the most frequent verdicts of possible errors.

Judgement Verdict	Description
Memory limit exceeded	The program tries to consume more memory than is indicated in the problem statement
Time limit exceeded	The program hadn't terminated in time indicated in the problem statement
Runtime error	The program terminated with a non-zero return code (possible reasons: array out of bound error, division by zero, stack overflow, incorrect pointers usage, etc)
Wrong answer	Wrong answer
Idleness limit exceeded	The program didn't use the CPU time for considerable time
Denial of judgement	The solution was impossible to run, perhaps, due to a judging error. The most probable cause is an error in the program (for example, using extra large arrays)

1. If the solution passed all pretests, the contestant is displayed the result that says **Pretests passed** . Otherwise, the contestant is displayed the judging result as the number of the first pretest that failed the judging and the judging system verdict.
2. If the result of the judging is **Compilation Error** , **Denial of judgement** (or similar) or if the solution didn't pass the first pretest, then this solution won't be considered in calculating results.
3. The solution of one problem can be submitted multiple times.
4. The solution that passes all pretests is considered the contestant's verified solution for this problem.
5. If a contestant submits several times a problem's solution that passes all pretests, then the last solution is considered as the contestant's verified solution for this problem. All other solutions will be considered as unsuccessful attempts.
6. If a contestant's solution passes all pretests, the problem is considered pre-solved by the contestant and the system calculates the contestant's preliminary points for the given problem. The points are calculated by the following scheme:
 - each minute decreases the problem's value: the value decreases by X/250 points per minute (where X is a problem's initial value), for example, a problem with the initial value of 500 points gets 2 points cheaper every minute;

- the number of points a contestant gets for a problem equals the current value of the problem in points minus penalty;
- the penalty is determined as the number of this contestant's previous solutions for this problem, multiplied by 50 points;
- a contestant can't earn less points than 30% of the initial problem value.

Hacks

1. A contestant can lock any of his pre-solved problems. That means that the contestant loses the right to submit solutions for this problem.
2. After the contestant locks a problem, he gets the right to view the original codes of submitted solutions of other contestants for this problem that share the same room with him. Having viewed another person's code, the contestant can suggest a test on which, as he thinks, the given solution will fail. This procedure will be further called a solution's hack. A test can be given manually or using a generator program that writes the test to the standard output. The system automatically validates the test to satisfy problem statement constraints. If it doesn't, the contestant is notified about it and the hack attempt is ignored. If the test satisfies the indicated constraints, then the solution runs on the offered test. If the solution doesn't pass the test, then the hack is considered successful, otherwise it is considered unsuccessful. A hack attempt is also ignored if by the given moment the hacked solution is not the contestants' last verified solution of the problem. For example, this might be the case if somebody already hacked the given solution before or the hacked contestant has resubmitted the solution.
3. A contestant gets 100 points for a successful hack and a penalty of 50 points for an unsuccessful hack.
4. If a contestant's solution is successfully hacked, then happens the following:
 - the problem isn't considered pre-solved by this contestant anymore
 - his preliminary points for this problem are reset to 0
 - the test from the hack is added to this problems' pretest set for the given contestant
5. If a contestant locked a problem, but had the solution hacked, then he preserves the right to hack other people's solutions for this problem.

Can-do's and Can't-do's

1. During the round the contestants are allowed to use any books and personal notes as well as a code written beforehand by you personally.
2. ~~It is forbidden to use somebody else's code in the solution.~~ You may use third-party prewritten code with some restrictions, read carefully [about it](#).
3. It is allowed to use any sources of information on the Internet (but it is forbidden to copy-paste somebody else's code).
4. It is forbidden to obfuscate the solution code as well as create obstacles for its reading and understanding. That is, it is forbidden to use any special techniques aimed at making the code difficult to read and understand the principle of its work.
5. The contestants are forbidden to talk about subjects, related to the problems, with anybody, including other contestants. It is only allowed to ask questions to the jury via the system (see the 'Questions' section).
6. The organizers of the contests have the right to monitor the contestants' honesty in behavior using different methods and disqualify the contestant if violations are found.
7. All programs submitted by the contestants for judging, should be aimed at solving the problem and not at violating rules or destabilizing the judging system. A contestant who deliberately attempts to destabilize the judging system, will be disqualified.
8. Attempting to digitally extract other contestant's code during the hacking is considered cheating. You may not use any technical/digital tools to obtain other contestant's code, including (but not limited) OCR, traffic capture, browsers plugins and so on. The only allowed method to analyze other contestant's solution is reading it in a hacking window. However it is allowed to manually retype the solution or it's parts to run it locally.
9. Any violation of these rules, rules from the FAQ section, rules from the contest announcement or rules that were accepted during the registration will lead to penalty provisions, including disqualification.

Standings

1. During the contest the current results' table is available to the contestants. The table shows the current place of each contestant. The table also contains the information about the hacks and the history of submissions of each contestant.
2. All intermediate results are unofficial.

System Testing and Final Standings

1. After a contest is over, the results of the contestants' solutions' judging on the main set of tests are published.
2. The main test set is used to judge verified solutions, i.e., the solutions that meet all these criteria:
 - passed all pretests;
 - are the last submitted solutions of each contestant for each problem;
 - haven't been hacked.
3. A problem is considered solved by the contestant if the contestant's solution passed all tests from the main set of tests.
4. A solved problem brings to the contestant the score, equal to the pre-score of his solution for this problem.
5. A contestant's result is the sum of points for all problems he has solved plus points for hacks.
6. In the final results' table the contestants are listed in the order of the results' decreasing. If two or more contestants have the same result, they share the same place.
7. The judging is performed on computers Core 2 Duo E6750, 2.66 Ghz, 3Gb of memory.

Intellectual property

1. The rules below are used each time if for a particular round there are no separate rules. Competitors retain ownership of all intellectual and industrial property rights (including moral rights) in and to Submissions.
2. As a condition of submission, Competitor grants Codeforces, its subsidiaries, agents and partner companies, a perpetual, irrevocable, worldwide, royalty-free, and non-exclusive license to use, reproduce, adapt, modify, publish, distribute, publicly perform, create a derivative work from, and publicly display the Submission.
3. Contestants provide submissions on an "as is" basis, without warranties or conditions of any kind, either express or implied, including, without limitation, any warranties or conditions of title, non-infringement, merchantability, or fitness for a particular purpose.

codeforces, rules

▲ +17 ▼ ☆

MikeMirzayanov

📅 13 years ago

💬 76



Comments (58)

☐ Show archived | [Write comment?](#)

13 years ago, # | ☆

▲ 0 ▼

Is it legal if I do following scenario?:



Przemek

- pass pretests and lock my solution
- download another user solution
- compile it (without reading source)
- run some tests on my machine
- hack solution (if for any test outputs are different)

→ [Reply](#)



Fefer_Ivan

13 years ago, # ^ | ☆

▲ +10 ▼

You can not download another user solution.

→ [Reply](#)