



What is the schedule of the event?

9:30AM - Meet up at contest site, announcements, set up compute environment.

10:00AM - Contest Starts

12:01PM - Pizza!

15:00PM - Contest Ends. Announce winning team(s), discuss solutions

15:30PM - Leave contest site.

What is the format of the contest?

Each team will have one computer to work on. When the contest starts, the organizers will provide you with printed copies of the contest problems. The problem statement text and will then also become available on open.kattis.com. Each problem describes a task, where you are to make a program that on given input (meeting prescribed specifications) should produce a given output (meeting prescribed specification), within the stated time limit (usually a few seconds). The input is read from stdin (e.g. `cin` in C++), output is written to stdout (e.g. `cout` in C++).

When you believe that you have a program that behaves according to specification, you may submit the *source code* of your program as a *single file* to the online judge. The judge will run your problem on a number of secret test cases (that all meet the input specifications). For each input it will check whether

your program (a) terminated within the time limit and (b) produced an output that matches the specifications (for that problem and that input instance). It will report a judgement quickly, typically within 30 seconds of submission.

1) If your submitted program does not compile, the online judge will report "compilation error".

2) If the program does not terminate within the time limit, the online judge will report "time limit exceeded".

3) If the program terminates, but fails to produce output that matches the specifications, the online judge will report "wrong answer".

4) If your program terminates within the time limit and produces correct output, the judge reports that your submission has been "Accepted". Congratulations, and move on to the next problem!

How are teams ranked?

Teams are ranked by the number of problems that they have gotten accepted (more is better). Teams who have solved the same number of problems are ranked according to total penalty time, computed as follows. The total penalty time of a team is the sum of the penalty times overall all problems that the team has gotten accepted. The penalty time of an accepted problem is the number of minutes since the beginning of the contest until the first accepted submission of that team for that problem. Additionally, each unsuccessful submission for that problem incurs a 20 minute penalty time.

Suppose the contest starts at 10:00am. A team first makes two unsuccessful submissions to problem A, then one unsuccessful submission to problem C, and then they get problem C accepted at 11:00am. Then they make three unsuccessful submissions to problem B, before getting problem A accepted at 12:00pm. Their penalty time for problem A is $(2 \times 20) + 120 = 160$ minutes. Their penalty time for problem B does not matter because they never got it accepted. Their penalty time for problem C is $20 + 60 = 80$ minutes. Their total penalty time is therefore 240 minutes.

Who gets invited to Regionals?

The top two teams consisting only of ICPC-eligible students (essentially all undergraduates are eligible, some beginning graduate students may be. See <https://icpc.global/regionals/rules>) will be invited. Additional teams and/or individual ICPC eligible students who participate in the UCSB contest may be invited at the discretion of the organizers.

Contest Rules:

- Each team may bring their own pens, scratch paper, and any written materials [books, printouts, etc.].
- Each team will have access to ONE computer (self brought laptop OR lab desktop).
- Contestants will be allowed to keep their phones, but during the period of the contest the smartphones should only be used for "practical" communication (e.g. "when will you be home for dinner?"). Using your phone or any other computing device for any other purpose is not allowed.
- Accessing the internet is only allowed to access open.kattis.com, read the problem statements, submit your solutions, and read the language documentations that are linked to from the help page on open.kattis.com.
- Discussing the problems with anyone other than the judges and your teammates is strictly prohibited.
- In this contest you may use any programming language that Kattis accepts (see their help pages). However, if you make it to the regionals, then there you will have to code in C, C++, Java, Python 3 or Kotlin.
- All submitted code should be your own, and programmed during the contest. The only exceptions: (1) the standard library of the programming language you use, (2) code provided by Kattis (such as Kattio.java). Using non-standard libraries is explicitly forbidden. If you use your own laptop you may not copy from or even look at any of the code that is there.

Tips and tricks:

- Organizer(s) will be on site to help you (with anything else than solving the problems). Ask questions if you have any.
- Your program is supposed to read input from standard input and write output to standard output. For Java and C++ there are known issues related to performance (essentially if the input file is 1MB or more) then Java's Scanner and C++ cin functions could be too slow. For Java this is fixed using the Kattio.java class provided by Kattis (see their help page), for C++ you may add the line

```
ios_base::sync_with_stdio(false);
```

at the beginning of your `main()` method to make input and output much faster. If you do this, make sure not to use `printf` and `scanf` since then things can go bad.

- Kattis will accept programs that on the secret test cases (that are formatted precisely according to the specifications) run within the time limit and produce output that is precisely according to the specification of the problem. Using uppercase letters instead of lowercase letters (or vice versa), or a trailing punctuation "." that should not be there WILL result in your program getting judged "wrong answer". Therefore *read the problem statement carefully!*
- Many of you will have the following problem: you think that your program is correct, but when you submit you get "wrong answer". This is not the servers fault, there is not a mistake in the problem formulation or the test cases. These tasks are very carefully vetted. The problem is always with your program. Read the problem statement carefully. Try to make test cases that make your programs fail. Try to explain to your team mates what each line of your code does and why, until you reach that one line that does not make sense even to you.

- Even though time complexity is not explicitly evaluated, the time complexity of your algorithm WILL affect how fast it runs on large instances. Use the provided time limits and input specifications to determine how fast of an algorithm you need to code. There is no need to over-achieve, you should always aim to program the simplest algorithm that will get accepted given the specifications. A useful but very rough estimate is that the server does about about 10^8 instructions per second.
- Especially if you are not familiar with tasks on this format, it is advisable that you try to get a couple of the "trivial" tasks on open.kattis.com accepted before the contest, such as the tasks "hello" or "timeloop".
- You can “redirect” input from a file to stdin. Suppose you made a program that takes as input 3 numbers and outputs their sum. You would normally run it like this:

```
$ ./program  
10 20 30
```

To get the output 60.

If you now make a file “input.txt” containing the line
10 20 30

And run

```
$ ./program < input.txt
```

You will also get the output 60. This is nice because you can run a test case without typing it by hand.