

# F-Talent Code 2023

## Rehearsal Programming Contest

### Table A

F-Talent 2023

October 01, 2023



### Problems

- A. Censorship for Children
- B. Intersection Areas of Rectangles
- C. Oriented Minimum Bounding Box
- E. The Point Of Each Player
- F. Player ranking

Do not open before the contest has started.

## Advice, hints, and general information

- The problems are **not** sorted by difficulty.
- Your solution programs must read input from *standard input* (e.g. System.in in Java or cin in C++) and write output to *standard output* (e.g. System.out in Java or cout in C++). For further details and examples, please refer to your administrator guide and Domjudge documentation.
- For information about which compiler flags and versions are used, please refer to your administrator guide. (Python 2.7.17, Oracle Java 1.8.0\_144, gcc 7.5.0 (C, C++ std14)).
- Your submissions will be run multiple times, on several different inputs. If your submission is incorrect, the error message you get will be the error exhibited on the first input on which you failed.
  - E.g., if your instance is prone to crash but also incorrect, your submission may be judged as either “Wrong Answer” or “Run Time Error”, depending on which is discovered first. The inputs for a problem will always be tested in the same order.
- If you think some problem is ambiguous or underspecified, you may ask the judges for a clarification request through the Domjudge system. The most likely response is “No comment, read problem statement”, indicating that the answer can be deduced by carefully reading the problem statement or by checking the sample test cases given in the problem, or that the answer to the question is simply irrelevant to solving the problem.
- In general we are lenient with small formatting errors in the output, in particular whitespace errors within reason, and upper/lower case errors are often (but not always) ignored. But not printing any spaces at all (e.g. missing the space in the string “1 2” so that it becomes “12”) is typically not accepted. The safest way to get accepted is to follow the output format exactly.
- For problems with floating point output, we only require that your output is correct up to some error tolerance. For example, if the problem requires the output to be within either absolute or relative error of  $10^{-4}$ , this means that
  - If the correct answer is 0.05, any answer between 0.0499 and .0501 will be accepted.
  - If the correct answer is 500, any answer between 499.95 and 500.05 will be accepted.

Any reasonable format for floating point numbers is acceptable. For instance, “17.000000”, “0.17e2”, and “17” are all acceptable ways of formatting the number 17. For the definition of reasonable, please use your common sense.

## A. Censorship for Children

**Time Limit: 3 seconds**

### Problem description

You are asked to join a project to build a forum for children. The children can talk about cartoons they like. They can post articles or comment on them. Unfortunately, bad guys can join this forum and post inappropriate comments. Your task is to build a module capable of moderating these comments. A simple solution applied in the game League of Legends is to censor all the bad words that appear in the dictionary.

For example, we have a dictionary that includes the words: *"fuck, shit, hell, bitch"*.

If a bad guy posts a comment containing the above words : *"Bitch, I will send you to hell to eat shit"*.

The words will be censored: *"B\*\*\*h, I will send you to h\*\*l to eat s\*\*t"*.

### Input:

1<sup>st</sup> line of the INPUT represents the dictionary. 2<sup>nd</sup> line stands for the comment of the user.

### Output:

Censored text

Check out some examples below:

Test case	INPUT	OUTPUT
1	Fuck shit hell bitch The movie like Shit	The movie like S**t
2	fuck shit hell bitch Bitch, I will send you to hell to FUCK shit	B**h, I will send you to h**l to F**K s**t
3	fuck shit hell bitch hey fucker it is not ok man	hey f**ker it is not ok man
4	fuck shit hell bitch I love you	I love you