Practice Round

May 26, 2020

Problems[1]

A	Sum	stdin/stdout
B	Letter b	stdin/stdout
C	Guess the Number	interactive
D	Happy Easter	output-only

[1] See the last page for general information about the problems.

Problem A
Sum

You are given two space-separated integers. The integers are not larger than 10^9 in absolute value. You have to print their sum in the format given in the sample test below.

Sample Input 1	Sample Output 1
`2 2`	`The suM! of 2 and 2 IS 4`

Sample Input 2	Sample Output 2
`1 4`	`The suM! of 1 and 4 IS 5`

Problem B
Letter b

Given a non-empty string composed of lowercase letters of the English alphabet. Your goal is to calculate the number of times the letter ‘b’ appears in the string. The given string has at most 1000 characters.

Sample Input 1	Sample Output 1
abracadabra	2

Sample Input 2	Sample Output 2
abbazabba	4

Problem C
Guess the Number

This is an interactive problem. Your program will interact with the one written by the jury using the standard input and output. The jury's program is given a hidden integer h between 1 and n ($1 \leq n \leq 10^6$). The goal of your program is to guess h in at most 100 attempts. You make attempts and the jury's program replies whether your guess is larger, smaller or equal to the given number.

The interaction protocol. First, your program needs to read the number n from the first line of the standard input. Then your program should print its guess attempt x in the standard output. x must be an integer between 1 and n. Then your program reads the jury's program's response from one line of the standard input. You may get one of the following inputs:

- -1 – this means that $h < x$, i.e., the hidden number is smaller than your guess;
- 1 – this means that $h > x$, i.e., the hidden number is larger;
- 0 – this means that the hidden number and your guess are equal.

When your program finds the hidden number or receives EOF (end-of-file) signal, it must quit its execution. Do not forget to flush after each output. See the last page for more details about interactive problems.

Limits. The number n is between 1 and 10^6, and your program must find the answer in at most 100 attempts.

Note, in the examples below, the empty lines are only for presentation. Your program must not produce them.

Jury's feedback 1	**Your attempts 1**
5 1 1 1 0	 1 2 3 4

Jury's feedback 2	**Your attempts 2**
10 1 -1 0	 5 7 6

Problem D
Happy Easter

Count the number of red, green and blue eggs in the following Easter page.

Input

There is no input for this problem, except for the link above. Here is an alternative link to the page.

Output

You should simply upload or paste the numbers in the online judge, in the format shown below.

Example output	Explanation

`r: 12`
`g: 20`
`b: 9`
	`This is not the correct answer.`
	`It is just an explanation of the`
	`output format, which you should follow.`

General information about interactive and stdin/stdout problems

- Time limit for all interactive or stdin/stdout problems is 2 seconds and memory limit is 128MB.
- All input/output should be done from standard input and output. You can find examples of standard input/output in common languages here.

General remarks about interactive problems

1. You must print a new line after each interaction;
2. You must flush the output stream after each interaction:
   - In C or C++: `fflush(stdout);`
   - In Java: `System.out.flush();`
   - In Python: `sys.stdout.flush()`
   - In C#: `Console.Out.Flush();`
3. If your program receives EOF (end-of-file) condition on the standard input, it *must* exit immediately with exit code 0. Failure to comply with this requirement may result in Time Limit Exceeded error instead of other verdicts AC/WA/PE.
4. Typical issues with interactive problems are
   - Wrong Answer – usually means that your program followed the interaction protocol but the answer or the intermediate steps are wrong.
   - (Wall) Time Limit Exceeded – this means that due to your program's execution, the interaction is not progressing. This can happen
     - if your program is expecting an input from the jury's program by mistake. Most commonly this happens when the jury's program has detected an error and quit, but your program is waiting for an input. In this case, you will get the TLE verdict, instead of the actual WA/PE error. See point (3) above;
     - if your program has not provided the necessary output for the jury's program to respond. Most often this is because you have not flushed the output stream. See point (2) above.
   - Presentation Error – usually means that your program did not follow the interaction protocol correctly and the jury's interacting protocol is not able to test it.
   - Runtime error – usually a mistake in your program that makes your program crash during the execution.