UCF Local Contest (Round 1B) — September 12, 2020

Judging Assistant for Contest

filename: `autojudge`
Difficulty Level: `Easy-Medium`
Time Limit: `5 seconds`

When people try competitive programming for the first time, it can be challenging for some to write code according to the constraints of the contest, even if they are already good coders! Of course, the best way to learn is by doing, and that is why many contests have a "practice" or warm-up session, which includes testing out the full process of submitting code and getting responses from the judges. Still, there are always some people who do not attend that session and then make time-wasting mistakes in the real contest, where it counts! For example, they print out prompts for inputs, or forget to return zero from their C program. This happens often enough that the judges need some programs to help them evaluate the submissions.

The Problem:

Given information about a programming contest problem and a submission for that problem, help the judges determine the best response.

The Input:

The first input line contains the designated "filename" for the contest problem which is a string of 1 to 20 lowercase letters. Recall that this filename does not include the extension (.c, .cpp, .java, .py).

The second input line contains the name of the submitted file, a string of 1 to 70 characters. This filename may include an extension, though the contestant might have used an invalid extension (e.g., .html, .pl, .rb, .asp, …). Note that this file name consists of characters and not just letters, e.g., the file name may be "`C:\My Documents\graph.py`".

The third input line contains three single-space-separated integers: r ($0 \le r \le 10$), indicating the return code of the submitted program, d ($1 \le d \le 10$), indicating the time limit in seconds allowed for a correct program to run, and e ($0 \le e \le 20$), indicating the elapsed time in seconds while the program was running. (Note that, in a real contest, it may not be possible to run a submitted program but that aspect is not included in this problem to simplify the problem.)

The fourth input line contains an integer, c ($1 \le c \le 10$), indicating the number of output lines produced by a correct program. The following c input lines provide the correct output; each line will contain zero to 70 characters and the first and last line(s) will contain at least one non-blank character.

The next input line contains an integer, t ($0 \le t \le 10$), indicating the number of output lines produced by the submitted program. The following t input lines provide the output produced by the submitted program; each line will contain zero to 70 characters. Note that, unlike the correct output, it is not guaranteed that the first and last line(s) of the submitted output will contain at least one non-blank character, i.e., the submitted output could be all blanks. Note also that, in a real contest, a submitted program may have far more output lines than expected and some output lines may far exceed the expected length but those aspects are not included in this problem to simplify the problem.

The Output:

If the submitted file name does not match (case-sensitively) the designated problem filename, or if it doesn't have one of the valid extensions (.c, .cpp, .java, or .py), print the message "`Compile Error`". Otherwise, if the return code of the program is not zero, print "`Run-Time Error`". Otherwise, if the elapsed time for the submitted program exceeds the time limit, print "`Time Limit Exceeded`". Otherwise, if the submitted program output does not match the correct output (line by line and character by character), print "`Wrong Answer`". Otherwise, print "`Correct`".

Sample Input	**Sample Output**

`triangle`
`MyTriangle.py`
`0 3 0`
`1`
`scalene`
`0`
	`Compile Error`

`graph`
`graph.py`
`5 3 3`
`1`
`done`
`1`
`done`
	`Run-Time Error`

`dust`
`dust.c`
`0 10 11`
`3`
`12cm of dust`
`3cm of dust`
`Impossible`
`1`
`Please enter shelf size:`
	`Time Limit Exceeded`

(more Sample Input/Output on the next page)

Sample Input	Sample Output
awesome awesome.java 0 5 5 1 Result = 100 2 Result = 100	Wrong Answer
awesome awesome.java 0 5 5 2 Result = 100 2 RESULT = 100	Wrong Answer
awesome awesome.java 0 5 5 1 Everything is awesome! 1 Everything is awesome!	Correct