2003 Mid-Atlantic Regional Programming Contest
Practice Round

Welcome to the practice round for the 2003 Programming Contest. Before you start the contest, please be aware of the following notes:

1. There is one (1) practice problem. Please submit solutions or request clarifications **for this problem only.** Unless you have a real question about the problem, please submit at most one clarification request, and at most two runs. It is important that everyone have a chance to see how the system works.

2. All solutions must read from standard input and write to standard output. In C this is scanf/printf, in C++ this is cin/cout, and in Java this is System.in/System.out. The judges will ignore all output sent to standard error. (You may wish to use standard error to output debugging information.) From your workstation you may test your program with an input file by redirecting input from a file:

   ```
   program < file.in
   ```

3. Solutions for problems submitted for judging are called runs. Each run will be judged. Runs for each particular problem will be judged in the order they are received. However, it is possible that runs for different problems may be judged out of order. For example, you may submit a run for B followed by a run for C, but receive the response for C first. DO NOT request clarifications on when a response will be returned. If you have not received a response for a run within 60 minutes of submitting it, **you may have a runner ask the site judge to determine the cause of the delay. Under no circumstances should you ever submit a clarification request about a submission for which you have not received a judgment.**

   The judges will respond to your submission with one of the following responses. In the event that more than one response is applicable, the judges may respond with any of the applicable responses.

Response	Explanation
Correct	Your submission has been judged correct.
Incorrect Output	Your submission generated output that is not correct.
Output Format Error	Your submission's output is not in the correct format, is misspelled, or did not produce all of the required output.
Excessive Output	Your submission generated output in addition to or instead of what is required.
Compilation Error	Your submission failed to compile.
Run-Time Error	Your submission experienced a run-time error.
Time-Limit Exceeded	Your submission did not solve the judges' test data within 30 seconds.

4. A team's score is based on the number of problems they solve and penalty points, which reflect the amount of time and incorrect submissions made before the problem is solved. For each problem solved correctly, penalty points are charged equal to the time at which the problem was solved plus 20 minutes for each incorrect submission. No penalty points are added for problems that are never solved. Teams are ranked first by the number of problems solved and second by the fewest penalty points.

5. This problem set contains sample input and output for each problem. However, you may be assured that the judges will test your submission against several other more complex datasets, which will not be revealed until after the contest. Your major challenge is designing other input sets for yourself so that you may fully test your program before submitting your run. Should you receive an incorrect judgment, you are advised to consider what other datasets you could design to further evaluate your program.

6. In the event that you feel a problem statement is ambiguous, you may request a clarification. Read the problem carefully before requesting a clarification. If the judges believe that the problem statement is sufficiently clear, you will receive the response, "The problem statement is sufficient, no clarification is necessary." If you receive this response, you should read the problem description more carefully. If you still feel there is an ambiguity, you will have to be more specific or descriptive of the ambiguity you have found. If the problem statement is ambiguous in specifying the correct output for particular input, please include that input data in the clarification request.

   Additionally, you may submit a clarification request asking for the correct output for input you provide. The judges will seek to respond to these requests with the correct output. These clarification requests will be answered only when no clarifications regarding ambiguity are pending. The judges reserve the right to suspend responding to these requests during the contest.

   If a clarification, including output for a given input, is issued during the contest, it will be broadcast to all teams.

7. The submission of abusive programs or clarification requests to the judges will be considered grounds for immediate disqualification.

8. Good luck, and HAVE FUN!!!

Practice Problem: A Simple Question of Chemistry

Your chemistry lab instructor is a very enthusiastic graduate student who clearly has forgotten what their undergraduate Chemistry 101 lab experience was like. Your instructor has come up with the brilliant idea that you will monitor the temperature of your mixture every minute for the entire lab. You will then plot the rate of change for the entire duration of the lab.

Being a promising computer scientist, you know you can automate part of this procedure, so you are writing a program you can run on your laptop during chemistry labs. (Laptops are only occasionally dissolved by the chemicals used in such labs.) You will write a program that will let you enter in each temperature as you observe it. The program will then calculate the difference between this temperature and the previous one, and print out the difference. Then you can feed this input into a simple graphing program and finish your plot before you leave the chemistry lab.

Input

The input is a series of temperatures, one per line, ranging from -10 to 200. The temperatures may be specified up to two decimal places. After the final observation, the number 999 will indicate the end of the input data stream. All data sets will have at least two temperature observations.

Output

Your program should output a series of differences between each temperature and the previous temperature. There is one fewer difference observed than the number of temperature observations (output nothing for the first temperature). Differences are always output to two decimal points, with no leading zeroes (except for the ones place for a number less than 1, such as 0.01) or spaces.

After the final output, print a line with “End of Output”

Example

Input:

```
10.0
12.05
30.25
20
999
```

Output:

```
2.05
18.20
-10.25
End of Output
```