Sample Contest

Problems

Time limit for all problems: 1 seconds[1]
Memory limit for all problems: 256MB

Problems are not ordered by difficulty.
Do not open before the contest has started.

[1] Unless otherwise specified in the problem.

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
Guess the Number

This is an interactive problem. Your program will interact with the one written by 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 $x < h$;
- 1 – this means that $x > h$;
- 0 – this means that the hidden number of your guess are equal.

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

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

General remarks about interactive problems.

- You must print a new line after each interaction;
- 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();`
- 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.
  - 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. Note that this may happen if your output does not satisfy the required upper/lower limits of numbers.
  - Wall Time Limit Exceeded – this means that your program is not following the protocol in such a way that the interaction is not progressing. This can happen if your program is expecting an input from the jury's program by mistake; or if your program has not provided the necessary output for the jury's program to respond. The latter can happen if you do not flush the output stream.
  - Runtime error – usually a mistake in your program that makes your program crash during the execution.