Where's Waldo?

Problem Name	whereswaldo
Time Limit	11 seconds
Memory Limit	1 gigabyte

There is a hidden permutation $P_0, P_1, ..., P_{N-1}$ of length N, which is guaranteed to be generated uniformly at random. The permutation contains the numbers $1, 2, 3, ..., N$ exactly once each, in some unknown order.

You can choose positions l and r, and ask questions of the form: "What is the sum of $P_l + P_{l+1} + \cdots + P_r$?"

Your task is to find the position of the 1 in P using as few questions as possible. You will be scored depending on the number of questions used.

Interaction

Your program should first read two integers on a single line, T and N. T is the number of rounds your program will be tested on, and N is the length of P.

After this come T rounds:

When a round begins you may start to ask questions. Print a line with "? a b" to ask about the sum of the numbers between positions a and b inclusive ($0 \leq a \leq b \leq N - 1$).

After each question your program should read an integer, the sum of the numbers in the interval.

Once you have found the position of the 1, print a line of the form "! i", where i is the index such that $P_i = 1$. After you have printed this the next round will begin.

Make sure to flush standard output after asking a question, or else your program might get judged as Time Limit Exceeded. In Python, `print()` flushes automatically. In C++, `cout << endl;` also flushes in addition to printing a newline; if using printf, use `fflush(stdout)`.

Constraints and Scoring

Your program will be tested against **a single test case, with** $N = T = 1000$. The permutation in each test is guaranteed to be **generated at random**.

If your solution guesses wrongly in any of the rounds, your submission will be judged as *Wrong Answer*.

Otherwise, a score will be calculated as follows:

$$\text{score} = \min\left(220 - \tfrac{M}{2500}, 100\right) \text{ points,}$$

where M is the number of questions your program asks in total over all T rounds.

The score will be rounded to the nearest integer. If the score becomes negative it will be treated as zero points.

Thus, if you use more than $550\,000$ questions you will receive 0 points, and if you use $300\,000$ or fewer questions you will receive 100 points. In between, your score grows linearly.

Testing Tool

To facilitate testing of your solution, we provide a simple tool that you can download. See "attachments" at the bottom of the kattis problem page. The tool is optional to use, and you are allowed to change it. Note that the official grader program on kattis is different from the testing tool.

Example usage (with T=1000, N=10):

For python programs, say `solution.py` (normally run as `pypy3 solution.py`):

```
python3 testing_tool.py pypy3 solution.py <<<"1000 10"
```

For C++ programs, first compile it (e.g. with `g++ -std=gnu++17 solution.cpp -o solution.out`) and then run:

```
python3 testing_tool.py ./solution.out <<<"1000 10"
```

Example

In the sample testcase, $T = 2$ and $N = 10$. For the first of these two rounds, say the hidden permutation is "`6 10 8 7 9 1 2 4 5 3`". The first question `? 0 9` asks for the sum of all numbers, which is indeed 55, and the second question `? 0 4` asks for $6 + 10 + 8 + 7 + 9 = 40$.

grader output	your output
2 10	
	? 0 9
55	
	? 0 4
40	
	? 5 5
1	
	! 5
	? 0 0
1	
	! 0