7. Lazy Sorting (`sort`)

1 sec 100 points

Teacher Laur organized a programming contest for his N students and now wants to award them prizes according to their results. The prizes were lined up on a shelf in correct order, but then Toots toppled the shelf with his horseplay. As the prizes are in identical boxes, the teacher put them back on the shelf in a random order.

The prizes have different weights and the teacher has balance scales that he can use to compare two boxes and find out which is intended for the student with the better result. However, each weighing takes time and the first M students have already lined up behind his door to receive their prizes…

To minimize the total time that students spend waiting for their prizes, the teacher wants to give each student their prize with as few weighings as possible. Write a program to help the teacher do that.

Interaction. This is an interactive task. When your program starts, the first line of input contains N, the number of prizes N $(1 \leq N \leq 100)$, and M, the number of students lined up $(1 \leq M \leq N)$. The prizes are numbered $1 \ldots N$ in their order on the shelf.

Then the grading system will give M queries to your program. The program must process and answer each query; then the grading system gives the next query. After answering the last query, your program should quit.

Each query is given as integer K $(1 \leq K \leq N)$, indicating the rank of the next student coming to collect their prize (there were no ties in the contest).

To process the query, your program may ask the teacher to perform a number of weighings. To request a weighing, the program should output a line of the form '`?` A B', where A and B are the numbers of two prize boxes $(1 \leq A \leq N,\ 1 \leq B \leq N,\ A \neq B)$. In response, the grading system will give the result of the weighing on the next line of input: '`<`' if the prize in the box A is for the student with the better rank, or '`>`' otherwise.

When the program has identified the box holding the prize intended for the student ranked K-th, it should output a line of the form '`!` C', where C is the number of the box $(1 \leq C \leq N)$; with that, the query is processed. If there are more queries, the grading system will issue the next one on the next input line. When the program has answered all the queries, it should quit.

Grading. In this task, a program that gives the wrong prize to any student will get 0 points for that test case. A program that gives the correct prizes to all students will get points according to how much time in total the students spent waiting for their prizes. Specifically, each weighing while there are p students waiting will give the program p penalty points and the program's score for a test case is

$$100 \cdot \min(0.1 + 0.9^{100 \cdot P/Q - 99}, 1)\ \%$$

of the value of the test case, where $Q = N \cdot M \cdot \log_2(N \cdot M)/2$ and P is the sum of penalty points the program got in the test case. You may assume that in all test cases, the orders of both the prizes and the students are chosen randomly.

Example.

```
Input             Output
3 2
1
                  ? 1 2
>
                  ? 2 3
<
                  ! 2
3
                  ? 1 3
>
                  ! 1
```

There are 3 prizes and 2 students waiting in this example. Suppose the order of the prizes is $3, 1, 2$; that is, the first box contains the prize for the 3rd place, the second box the prize of the 1st place, and the third box the prize for the 2nd place.

The student who got the 1st place enters first. The program asks the teacher to weigh the boxes 1 and 2 and gets the response that the box 2 contains the better prize. Then the program asks the teacher to weigh the boxes 2 and 3 and gets the response that again the box 2 contains the better prize. Thus, the prize for the 1st place must be in the box 2 and the program outputs this as the answer.

The student who got the 3rd place enters next. The program asks the teacher to weigh the boxes 1 and 3 and gets the response that in this pair, the box 3 contains the better prize. Thus, the prize for the 3rd place must be in the box 1 and the program outputs this as the answer.

With that, all the queries are processed and the program quites, having accumulated $2 \cdot 2 + 1 \cdot 1 = 5$ penalty points (2 weighings with 2 students waiting and 1 weighing with 1 student waiting); thus the program scores 100 % in this test case.

Remark. To ensure your output is delivered to the grading system, you have to flush the output buffer after each line:

Language	Command

C	`fprintf(stdout, ...);` or `printf(...);`
	`fflush(stdout);`

C++	`cout << ... << endl;`

Java	`System.out.println(...);`
	`System.out.flush();`

Python	`sys.stdout.write(...)` or `print(...)`
	`sys.stdout.flush()`