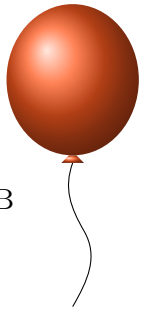


E Game: Cursor Minimum

TIME LIMIT: 2.0s
 MEMORY LIMIT: 1024MB

This is an interactive problem.

A permutation p of length $n = 30000$ is hidden by the judge. Your task is to find the index of the minimum value of this permutation.

The judge is non-adaptive: the permutation is chosen before the interaction starts and never changes.

Unlike ordinary comparison queries, the judge only compares your current query with your previous query. More precisely, if your previous queried index was i and now you query index j , the judge tells you the comparison between p_i and p_j .

You may make at most $q_{\max} = 42000$ queries.

The official judging data consists of 100 separate test files. Your program is run independently on each file.

INTERACTION

At the beginning of the interaction, your program must read two integers

$$n \quad q_{\max}.$$

For official tests, $n = 30000$ and $q_{\max} = 42000$.

To make a query, print one line in the following format:

$$? \quad j$$

where $1 \leq j \leq n$.

The judge replies with one character:

- $<$ if $p_i < p_j$ where i denotes the previous query (if applicable);
- $=$ if this is the first query, or if $p_i = p_j$;
- $>$ if $p_i > p_j$ where i denotes the previous query (if applicable).

Since p is a permutation, the reply $=$ after the first query can only happen if you query the same index as in the previous query. If the judge replies with -1 , your program must terminate immediately. This means that your program has violated the interaction protocol and will receive Wrong Answer.

To give your final answer, print one line in the following format:

$$! \quad a$$

where a is the index that you claim contains the minimum value. After printing the answer, your program must terminate. **The final answer is not counted as a query.**

If your program makes more than q_{\max} queries, queries an invalid index, prints an invalid command, or outputs an incorrect final answer, it receives Wrong Answer.

After every printed query or answer, flush the output. For example, in C++ you may use `endl` or `cout.flush()`.

SAMPLES

Sample input 1	Sample output 1
<pre>4 5 (receiving participant's output) = (receiving participant's output) > (receiving participant's output) < (receiving participant's output)</pre>	<pre>(receiving jury's output) ? 1 (receiving jury's output) ? 2 (receiving jury's output) ? 4 (receiving jury's output) ! 2</pre>

Explanation of sample 1.

This sample is only an example interaction and will not appear in the real test data. The lines of the form `(receiving ... output)` are placeholders used only to align queries and judge replies in the sample. In a real test case, the jury will not output these placeholder lines, and the participant should neither read nor print them.

In this example interaction, the hidden permutation is $p = (3, 1, 4, 2)$. The bullet points below describe the interaction shown in the sample.

- The judge first sends $n = 4$ and query limit 5.
- The first query `? 1` receives `=` because there is no previous queried index.
- The query `? 2` compares $p_1 = 3$ with $p_2 = 1$, so the judge replies `>`.
- The query `? 4` compares $p_2 = 1$ with $p_4 = 2$, so the judge replies `<`. The minimum is at index 2, so `! 2` is correct.

Local interaction tool: The attached `attachments/local_interactive.py` can reproduce this local setting with `--perm 3,1,4,2 --limit 5`, but passing it does not guarantee passing the official judging data.