

Problem C. Small Numbers Search

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

This is an interactive problem.

Jury has a permutation of numbers from 1 to n . Your task is find positions where numbers from 1 to k are placed. To do this, you can use jury's program which can compare numbers in any two positions in the permutation.

Input

The first line of input contains two integers n and k : the order of permutation and the number of positions to find. In all tests except the example, $n = 10\,000$ and $k \leq 10$.

Then follow the answers for your requests, one per line. If the first of the two numbers to compare is less than the second one, the line will contain a single character "<", otherwise, it will contain a single character ">".

Output

If you want to compare numbers on positions i and j , you must print one line "? i j ". Here, i and j must be different integers between 1 and n . You can request a comparison at most 10 700 times.

If you found all positions for all numbers from 1 to k , print "! pos_1 pos_2 ... pos_k ", and then terminate your program.

To prevent output buffering, after printing each line, consider issuing the command which flushes the buffer. For example, this command may be `fflush (stdout)` in C or C++, `System.out.flush ()` in Java, `flush (output)` in Pascal or `sys.stdout.flush ()` in Python.

Also, don't forget to put a newline at the end of every line of your output.

Example

standard input	standard output
3 3	<i>(reading input)</i>
<i>(waiting for output)</i>	? 1 2
<	<i>(reading input)</i>
<i>(waiting for output)</i>	? 3 1
>	<i>(reading input)</i>
<i>(waiting for output)</i>	? 2 3
<	<i>(reading input)</i>
	! 1 2 3
	<i>(terminating)</i>

Explanation

In the example, the jury's permutation is 1 2 3.