

BPC 1 S2 - Train Commute

Time limit: 2.0s **Memory limit:** 256M

You have moved to a new city, called Bytetown. The citizens of Bytetown get around by using one central rail line. This rail line is straight, and positions on it are numbered from 0 to C . There are N stations on it at integer positions. It is guaranteed that all stations are at least 2 units apart from each other. You can get on or off the train at any of the stations; then, you will be transported to any other station of your choice. You can also walk parallel to the train line.

However, you don't know where the stations are, and your friend, who has been living in Bytetown for a long time, is not willing to tell you. Instead, he allows you to make queries of the form "what is the minimum amount of distance one needs to walk to go from position a to position b , assuming they can use the train as much as they want?". You want to figure out the positions of the stations while using at most Q queries.

Constraints

$$2 \leq N \leq 10\,000$$

$$2 \leq C \leq 10^9$$

$$Q = 32\,768$$

Subtask 1 [15%]

$$N \leq 1\,000$$

Subtask 2 [85%]

No additional constraints.

Interaction

This is an interactive problem.

The first line contains three integers N , C , and Q .

After that, you can make a query by outputting `? a b`, where $0 \leq a, b \leq C$.

The grader will respond with a single integer representing the minimum amount of distance one needs to walk to go from position a to b , or `-1` if you made an invalid query or exceeded the query limit. You should terminate your program immediately after receiving a `-1` response to receive the correct verdict.

You should output `! p_1 p_2 p_3 ... p_n` once you believe you have determined the positions of the stations, where p_i is the position of the i^{th} station. You may output the positions in any order. You should make this query exactly once at the end of your program. This query does not count towards the query limit.

You should flush the output stream after every query. This can be done with `fflush(stdout);` or `std::cout << std::flush;` in C++, `System.out.flush();` in Java, or `sys.stdout.flush()` in Python.

You will receive a verdict of `Presentation Error` if your output has incorrectly formatted queries (the wrong number of tokens, incorrect whitespace, no `!` type query, etc.) and `Wrong Answer` if you exceed the query limit, guess the positions incorrectly, or output integers outside the specified ranges.

After outputting `-1`, the grader will not respond to any further queries, and therefore you should terminate your program to avoid receiving a `Time Limit Exceeded` verdict.

Sample Interaction

`>>>` denotes your output. Do not print this out.

The positions of the stations are `[0, 2, 5]`.

```
3 10 32768
>>> ? 1 6
2
>>> ? 0 5
0
>>> ? 3 4
1
>>> ? 5 10
5
>>> ! 5 0 2
```