C1. Guessing the Greatest (easy version)

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

The only difference between the easy and the hard version is the limit to the number of queries.

This is an interactive problem.

There is an array a of n **different** numbers. In one query you can ask the position of the second maximum element in a subsegment $a[l..r]$. Find the position of the maximum element in the array in no more than **40** queries.

A subsegment $a[l..r]$ is all the elements $a_l, a_{l+1}, \ldots, a_r$. After asking this subsegment you will be given the position of the second maximum from this subsegment **in the whole** array.

Input

The first line contains a single integer n ($2 \le n \le 10^5$) — the number of elements in the array.

Interaction

You can ask queries by printing "? l r" ($1 \le l < r \le n$). The answer is the index of the second maximum of all elements $a_l, a_{l+1}, \ldots, a_r$. Array a is fixed beforehand and can't be changed in time of interaction.

You can output the answer by printing "! p", where p is the index of the maximum element in the array.

You can ask no more than **40** queries. **Printing the answer doesn't count as a query**.

After printing a query do not forget to output end of line and flush the output. Otherwise, you will get `Idleness limit exceeded`. To do this, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `flush(output)` in Pascal;
- `stdout.flush()` in Python;
- see documentation for other languages

Hacks

To make a hack, use the following test format.

In the first line output a single integer n ($2 \le n \le 10^5$). In the second line output a permutation of n integers 1 to n. The position of n in the permutation is the position of the maximum

Example

input	Copy
5 3 4	

output Copy

```
? 1 5

? 4 5

! 1
```

Note

In the sample suppose a is $[5, 1, 4, 2, 3]$. So after asking the $[1..5]$ subsegment 4 is second to max value, and it's position is 3. After asking the $[4..5]$ subsegment 2 is second to max value and it's position in the whole array is 4.

Note that there are other arrays a that would produce the same interaction, and the answer for them might be different. Example output is given in purpose of understanding the interaction.
