The least missing element of a sorted array

Cost: 7 | Solved: 89

Memory limit: 256 MBs

Time limit: 1 s

Input: standard input

Output: standard output

Task:

This is an interactive task.

For this task, all your input is not premade but is created during the process of testing your program (which means you *can't* simply input all data at the beginning). For this purpose, we'll be using a special program called "the interactor", whose output will be sent to your solution and then vice versa. In other words, your solution and the interactor will be working in pair, and both your program and the interactor will decide what to output at a certain moment of time based on the history of their "conversation".

There's no way to input the array as you'd normally do. Instead, to get its elements, you have to send queries to the interactor. Queries are divided into two types:

? i – a query that makes the interactor return two values: function value and value of its derivative at the point *i*.

! i – a query that you have to send only at the end of the program, after you've found the required element.

To transfer data correctly, the interactor needs to use an operation *flush*, which most likely already exists in the standard library of your programming environment. For example, in C++ you can use *fflush(stdout)* or *cout.flush()* (depending on which one you're using – *scanf/printf* or *cin/cout*). In Java you can use method *flush* in the output stream, for example, *System.out.flush()*. In Python you can use *stdout.flush()*. In Pascal you can use *flush(output)*. Every data output must end with a line feed.

In interactive tasks inputting/outputting works way slower than normally, thus use *scanf/printf* instead of *cin/cout* in C++, *BufferedReader/PrintWriter* in Java and so on.

So, for example, if we have an array like this { 2, 8, 7, 4, 1 } and want to find the element that has a value of k = 7, we'd need to do the following (not an actual algorithm, just an example of how the interaction should happen):

```
#include <iostream>

using namespace std;

int main()

{

cout.flush();            //starting the interactor

int n, k, index, value;

cin >> n >> k;           //inputting the array size and the value we need to find

index = 3;               //the position we assume our element k is at

cout << "? " << index;   //asking the interactor what value the element at the position [in
                         //The interactor will look into the array and return you the value

cin >> value;            //inputting the returned value of the element

if (value == k)          //if this condition is true, it means we have found the element we
{
    cout << "! " << index << endl;  //telling the interactor that the required element is a
}

}
```

You are given a sorted array of ***n*** natural numbers.

Find the least missing element of this array.

For example, in the array 0 1 3 5 the least missing element is 2.

The quantity of queries cannot exceed 50.

Input:

The interactor inputs a number ***n*** *(**n** <= 10^9)* **–** the quantity of elements of the array.

Output:

The output should contain correct queries and the correct answer at the end. Every query should be output on a new line.

Example:

Input	Output

5 0 1 3	? 1 ? 2 ? 3 ! 2