

## D. Interactive LowerBound

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

*This is an interactive problem.*

You are given a **sorted** in increasing order singly linked list. You should find the minimum integer in the list which is greater than or equal to  $x$ .

More formally, there is a singly linked list built on an array of  $n$  elements. Element with index  $i$  contains two integers:  $value_i$  is the integer value in this element, and  $next_i$  that is the index of the next element of the singly linked list (or  $-1$ , if the current element is the last). The list is sorted, i.e. if  $next_i \neq -1$ , then  $value_{next_i} > value_i$ .

You are given the number of elements in the list  $n$ , the index of the first element  $start$ , and the integer  $x$ .

You can make up to 2000 queries of the following two types:

- `? i` ( $1 \leq i \leq n$ ) — ask the values  $value_i$  and  $next_i$ ,
- `! ans` — give the answer for the problem: the minimum integer, greater than or equal to  $x$ , or `-1`, if there are no such integers. Your program should terminate after this query.

Write a program that solves this problem.

### Input

The first line contains three integers  $n$ ,  $start$ ,  $x$  ( $1 \leq n \leq 50000$ ,  $1 \leq start \leq n$ ,  $0 \leq x \leq 10^9$ ) — the number of elements in the list, the index of the first element and the integer  $x$ .

### Output

To print the answer for the problem, print `! ans`, where `ans` is the minimum integer in the list greater than or equal to  $x$ , or `-1`, if there is no such integer.

### Interaction

To make a query of the first type, print `? i` ( $1 \leq i \leq n$ ), where `i` is the index of element you want to know information about.

After each query of type `? i` read two integers  $value_i$  and  $next_i$  ( $0 \leq value_i \leq 10^9$ ,  $-1 \leq next_i \leq n$ ,  $next_i \neq 0$ ).

It is guaranteed that if  $next_i \neq -1$ , then  $value_{next_i} > value_i$ , and that the array values give a valid singly linked list with  $start$  being the first element.

Note that you can't ask more than 1999 queries of the type `? i`.

If  $next_i = -1$  and  $value_i = -1$ , then it means that you asked more queries than allowed, or asked an invalid query. Your program should immediately terminate (for example, by calling `exit(0)`). You will receive "Wrong Answer", it means that you asked more queries than allowed, or asked an invalid query. If you ignore this, you can get other verdicts since your program will continue to read from a closed stream.

Your solution will get "Idleness Limit Exceeded", if you don't print anything or forget to `flush` the output, including the final answer.

To `flush` you can use (just after printing a query and line end):

- `fflush(stdout)` in C++;
- `System.out.flush()` in Java;

- `stdout.flush()` in Python;
- `flush(output)` in Pascal;
- For other languages see documentation.

## Hacks format

For hacks, use the following format:

In the first line print three integers  $n$ ,  $start$ ,  $x$  ( $1 \leq n \leq 50000$ ,  $1 \leq start \leq n$ ,  $0 \leq x \leq 10^9$ ).

In the next  $n$  lines print the description of the elements of the list: in the  $i$ -th line print two integers  $value_i$  and  $next_i$  ( $0 \leq value_i \leq 10^9$ ,  $-1 \leq next_i \leq n$ ,  $next_i \neq 0$ ).

The printed structure should be a valid singly linked list. In particular, it should be possible to reach all elements from  $start$  by following links  $next_i$ , and the last element  $end$  should have  $-1$  in the  $next_{end}$ .

## Example

input
<pre> 5 3 80 97 -1 58 5 16 2 81 1 79 4 </pre>
output
<pre> ? 1 ? 2 ? 3 ? 4 ? 5 ! 81 </pre>

## Note

You can read more about singly linked list by the following link: [https://en.wikipedia.org/wiki/Linked\\_list#Singly\\_linked\\_list](https://en.wikipedia.org/wiki/Linked_list#Singly_linked_list)

The illustration for the first sample case. Start and finish elements are marked dark.