Scroll of Secrets

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

This is an interactive problem

In the vast kingdom of DataLand, the wise King Algorithmus has hidden a series of magical scrolls along a secret path. Each scroll contains a number, and the scrolls are arranged in strict order, with each successive scroll bearing a larger number than the last. The kingdom's scholars refer to this path as a *singly linked list*, where each scroll not only holds a number but also points to the next scroll in the sequence.

You, the kingdom's chief seeker, have been tasked with finding the first scroll whose number is not less than a magical number x, which the king has entrusted to you. The challenge lies in the fact that the path is long, and you may only inspect a limited number of scrolls before the path vanishes into the mists of time.

The King hands you the magical scroll

Your mission is to navigate this mysterious path efficiently. Starting from a given scroll, you must determine the smallest number that is greater than or equal to x. If no such number exists, you must return to the king with the news that the quest has failed.

The Path:

- The path consists of n scrolls, each containing a number and a clue (an index) pointing to the next scroll.
- You begin your quest at a scroll indexed as `start`.
- The scrolls are sorted in ascending order, meaning the number on each scroll is less than the number on the next scroll, as long as it exists.

Your Tools:

- You can ask about a specific scroll by its index to reveal the number it holds and the index of the next scroll.
- However, the ancient magic of the path only allows you to ask up to 5000 questions before the path dissolves.

You task is to find and report the smallest number on the path that is greater than or equal to x. If no such number exists, report that the quest has failed.

Input

The number of scrolls n, the index of the first scroll `start`, and the magical number x.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq \texttt{start} \leq n$
- $0 \leq x \leq 10^9$
- You may ask no more than 5000 questions.

Output

- If you find the desired number, return it to the king. Otherwise, return -1 to signify the quest's failure.

Interaction Protocol

- To make a query about a scroll, print `? i`, where i is the index of the scroll.
- After printing, read two values: `value_i`, which is the number on the scroll, and `next_i`, which is the index of the next scroll in the sequence.
- You may make up to 5000 such queries.
- To submit your final answer, print `! ans`, where `ans` is the smallest number greater than or equal to x. If no such number exists, print `! -1`.
- After the final answer is submitted, your program should terminate.

Scoring

The scoring for this problem depends on the number of queries used:

- If you use $\leq$ 100000 queries, you will receive 50% of the points.
- If you use $\leq$ 60000 queries, you will receive 75% of the points.
- If you successfully use $\leq$ 5000 queries, you will receive full points.

Example

standard input	standard output
5 3 80 97 -1 58 5 16 2 81 1 79 4	81

Note

For the first example, in a quest where $n = 5$, `start = 3`, and $x = 80$, your path may look something like this:

- Scroll 1 holds the number 97 and is the last scroll.
- Scroll 2 holds the number 58 and points to Scroll 5.
- Scroll 3 holds the number 16 and points to Scroll 2.
- Scroll 4 holds the number 81 and points to Scroll 1.
- Scroll 5 holds the number 79 and points to Scroll 4.

A strategy would be to ask 5 queries to get the information for all scrolls, then you would report back to the king with the number 81 as it is the smallest number greater than or equal to 80.

Hints:

- Think about the *nuts and bolts* problem discussed in class. Just as matching each nut to its corresponding bolt required clever strategies, this problem may benefit from a similar approach. Consider using randomization to explore the scrolls efficiently and find the correct number. The key lies in making wise choices about which scrolls to inspect.