Problem H. Yurik and Important Tasks

Time limit: 1 second
Memory limit: 512 megabytes

Yurik is a very busy person, so he doesn't have time to complete all the necessary tasks on time. By the end of the week, he has accumulated n unfinished tasks. For convenience, let's number them using integers from 1 to n. At first, Yurik decided to do the tasks in the order of their numbering. However, he quickly realized that some tasks are more important than others, so he decided to change the order of their execution.

Yurik has a favorite *permutation* $p_1, p_2, \ldots, p_n$, and he decided to use it to change the order of task execution. Besides being very busy, Yurik is also very fickle, so he decided to change the order of tasks q times.

Initially, Yurik wrote down the tasks in the order he would perform them on a piece of paper: $1, 2, 3, \ldots, n$. After that he will choose two numbers l and r ($1 \le l \le r \le n$) q times, and then change the order of the tasks that are currently in the list at positions from l to r.

The change in the order of task execution happens as follows. First, Yurik assigns *priorities* to the tasks that are in the list at positions from l to r. The task at position l will be assigned priority p_1, the task at position $l + 1$ will be assigned priority p_2, and so on. Thus, the task at position r will be assigned priority p_{r-l+1}. After that Yurik rearranges the tasks in **ascending** order of their priorities. For a better understanding of the process, pay attention to the explanation in the examples.

After Yurik changes the order of task execution q times, he gets confused and now wants to know which task he will perform as the k-th in order. Help him figure this out.

Recall that a permutation $p_1, p_2, \ldots, p_n$ is an array of pairwise distinct integers from 1 to n.

Input

The first line contains three integers n, q, and k ($1 \le n \le 100\,000$; $1 \le q \le 100\,000$; $1 \le k \le n$).

The second line contains n integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le n$) — Yurik's favorite permutation.

Each of the following q lines contains two integers l_i and r_i ($1 \le l_i \le r_i \le n$) — the start and end of the range of tasks that Yurik will change the order of in the i-th time.

Output

Print a single integer t ($1 \le t \le n$) — the number of the task that Yurik will perform as the k-th in order after all the changes in the order of task execution.

Example

standard input	standard output
8 6 3 1 5 2 8 7 4 3 6 1 8 2 4 7 8 1 1 1 3 2 7	7

Note

Let's consider the first example. Initially, Yurik planned to perform the tasks in the following order: $1, 2, 3, 4, 5, 6, 7, 8$. However, he decided to change the order of task execution 6 times.

In the first time, Yurik chose the tasks that are in the list at positions from 1 to 8 (i.e., all the tasks he had). The first task was assigned priority 1, the second task was assigned priority 5, the third task was assigned priority 2, and so on. After that, Yurik rearranged the tasks in ascending order of their priorities, resulting in the sequence: $1, 3, 7, 6, 2, 8, 5, 4$.

In the second time, Yurik chose the tasks that are in the list at positions from 2 to 4: $3, 7$, and 6. The task with number 3 was assigned priority 1, the task with number 7 was assigned priority 5, and the task with number 6 was assigned priority 2. After Yurik rearranged the tasks in ascending order of their priorities, the sequence became $1, 3, 6, 7, 2, 8, 5, 4$.

In the third time, Yurik chose the last two tasks in the current order: 5 and 4. The first task was assigned priority 1, and the second task was assigned priority 5, so their order of execution did not change.

In the fourth time, only the first task was chosen.

In the fifth time, the first three tasks in the current order were chosen, and after assigning priorities, tasks 3 and 6 swapped places.

Finally, in the sixth time, all tasks except the first and the last were chosen. The final order of task execution looks like this: $1, 6, 7, 5, 3, 8, 2, 4$.