2018 Canadian Computing Olympiad
Day 2, Problem 1
Gradient Descent

Time Limit: 1 second

Problem Description
Troy wants to play the following game with you.

He has a grid with R rows and C columns. Rows are numbered from 1 to R and columns are numbered from 1 to C. Let the cell at row p and column q be denoted as (p, q).

There are N tokens numbered from 1 to N. Token i is placed at (X_i, Y_i) where $1 \leq X_i \leq R$ and $1 \leq Y_i \leq C$. There may be multiple tokens at the same cell. In one second, Troy can move one token to a horizontally or vertically adjacent cell. The *score* of a cell is defined as the minimum number of seconds needed for Troy to move every token to this cell.

Your goal is to find the minimum score of any cell in the grid. Unfortunately, Troy does not tell you how many tokens there are or where they are placed. However, you may ask him questions. You can ask Troy to tell you the score of any cell (p, q). You may ask at most K questions before Troy gets bored.

Interaction Protocol
This problem is interactive: input will be given based on questions generated by your program.

First, read one line with three integers R, C, K $(1 \leq R, C \leq 10\,000\,000; 1 \leq K \leq 170)$.

After your program has read this line, your program may ask questions.

To ask a question about (p, q) $(1 \leq p \leq R; 1 \leq q \leq C)$, print one line in the format "? p q". Then, read one line with one integer s $(0 \leq s \leq 2\,000\,000\,000)$, the score of (p, q).

Once your program determines the minimum score is Z, print one line in the format "! Z". Your program must terminate immediately after printing this line.

The output must be flushed after every line is printed, including the last line. To flush you can use: `fflush(stdout)` or `cout << endl` in C/C++; `System.out.flush()` in Java; `flush(output)` in Pascal.

If any printed line is wrongly formatted or you ask more than K questions you will get an *incorrect* verdict.

For every test case, the grading system will have fixed integer values N, R, C, K, X_1, ..., X_N, Y_1, ..., Y_N $(1 \leq N \leq 100; 1 \leq X_i \leq R; 1 \leq Y_i \leq C)$. These values will remain constant while your program is running. That is, the grading system is **not adaptive**.

For 5 of the 25 available marks, $R = 1$, $C \leq 90$ and $K = 90$.

For an additional 5 of the 25 available marks, $R = 1$, and $K = 90$.

For an additional 5 of the 25 available marks, $K = 170$.

For an additional 5 of the 25 available marks, $K = 100$.

For the remaining 5 marks, $K = 75$.

Sample Interaction 1

Request to grader	Feedback from Grader
	`1 10 90`
`? 1 3`	`9`
`? 1 7`	`11`
`? 1 4`	`8`
`! 8`	

Explanation of Output for Sample Input 1
This sample corresponds to tokens at cells $(1, 2)$, $(1, 4)$ and $(1, 10)$. It is guaranteed this sample will match the sample test in the grader.

The score of cell $(1, 3)$ is $1 + 1 + 7 = 9$.

The score of cell $(1, 7)$ is $5 + 3 + 3 = 11$.

The score of cell $(1, 4)$ is $2 + 0 + 6 = 8$ and this is the minimum in the grid.

For information, here are the scores in this example:

Column	1	2	3	4	5	6	7	8	9	10
Score	13	10	9	8	9	10	11	12	13	14

Sample Interaction 2

Request to grader	Feedback from Grader
	`5 4 170`
`? 2 4`	`11`
`? 1 4`	`15`
`? 3 3`	`7`
`! 7`	

Explanation of Output for Sample Input 2
This sample corresponds to tokens at cells $(2, 3)$, $(2, 3)$, $(4, 3)$ and $(5, 1)$.

2018 Canadian Computing Olympiad
Day 2, Problem 2
Boring Lectures

Time Limit: 8 seconds

Problem Description

You have a schedule of N upcoming lectures that you have the option of attending. The lectures are numbered from 1 to N and are in chronological order. From the current schedule, you expect that the ith lecture will have quality a_i. Since most of the lectures will be boring, you are only willing to attend some group of K consecutive lectures. You will skip the remaining lectures so that you can catch up on sleep and participate in programming contests. Since you don't like taking notes, you will only be able to remember the content from 2 of the lectures you attend. You want to choose the lectures you attend and the 2 lectures you remember as to maximize the sum of the lecture qualities of those 2 lectures.

There are Q changes that will be made to the schedule. The jth change to the schedule is represented by two values i_j, x_j that indicate that the quality of the i_jth lecture changes to x_j. For each of the $Q + 1$ versions of the schedule, find the maximum possible sum of lecture qualities that you can attain.

Input Specification

The first line will contain three integers N, K, and Q ($2 \leq N \leq 10^6$, $2 \leq K \leq N$, $0 \leq Q \leq 10^5$). The second line contains N integers $a_1, \ldots, a_N$, ($0 \leq a_i \leq 10^9$). The next Q lines each contain two integers i_j and x_j ($1 \leq i_j \leq N$, $0 \leq x_j \leq 10^9$).

For 5 of the 25 available marks, $Q = 0$.

For an additional 10 of the 25 available marks, $N \leq 10^4$.

Output Specification

Output $Q + 1$ lines, each containing a single integer. The j-th line that follows should contain the answer for the schedule obtained after the first $j - 1$ changes are made.

Sample Input

```
4 3 1
6 1 2 4
1 3
```

Output for Sample Input

```
8
6
```

Explanation for Output for Sample Input
For the original schedule, it is best to attend the first three lectures and remember the first and third, for an overall value of $6 + 2 = 8$. After the update, it is best to attend the last three lectures and remember the last two, giving a value of $2 + 4 = 6$.

2018 Canadian Computing Olympiad
Day 2, Problem 3
Flop Sorting

Time Limit: 2 seconds

Problem Description
Desperate to contribute to the CCO, Robert tried inventing a segment tree problem. The specification for that problem is:

> You are given a list of N distinct integers between 1 and N. These are arranged in a row such that the i-th integer from the left is a_i, where $1 \leq i \leq N$. We define a ***flop*** operation on a set of elements as swapping the minimum element of the set with the maximum element of the set. There will be Q flop operations, each specifying two numbers l and r ($1 \leq l \leq r \leq N$). For each operation, you must perform a ***flop*** on the interval $[l, r]$ (that is, on the segment of numbers $a_l, a_{l+1}, \ldots, a_{r-1}, a_r$). You must perform the Q flops in the order given, and report the final result.

Now that he is finished with the problem statement, Robert needs to create some test data. For one test case in particular, he is trying to encode an inside joke into the initial and final sequences of numbers. With these two sequences fixed, help him find any sequence of flop operations that transforms the first sequence into the second.

Input Specification
The first line will contain the integer N ($1 \leq N \leq 4096$). The second line will contain N distinct space separated integers between 1 and N, representing the initial sequence. The third line will also contain N distinct space separated integers between 1 and N, representing the final sequence.

Output Specification
The first line of output should contain the integer Q, with $Q \leq 300\,000$. The next Q lines should contain two integers l and r with $1 \leq l \leq r \leq N$.

For 5 of the available 25 marks, $N \leq 100$.

For an additional 10 of the available 25 marks, $N \leq 2048$.

Sample Input
```
6
1 3 5 6 4 2
1 2 3 4 5 6
```

Output for Sample Input

```
4
2 3
3 6
2 5
4 5
```

Explanation for Output for Sample Input

The first flop swaps 3 and 5, the second flop swaps 6 and 2, the third flop swaps 5 and 2, and the fourth flop swaps 5 and 4.