Task 4: `Robots`

Funnyland is modelled as a grid of size $(h+2) \times (w+2)$. The rows of the grid are numbered 0 to $h+1$ from top to bottom, and the columns of the grid are numbered 0 to $w+1$ from left to right. We refer to the cell located at row r and column c of the grid as cell (r, c).

Initially, all cells in this grid are coloured *white*, except for the rightmost column of cells, which are all coloured *black*.

The columns 1 to w each contain exactly one special cell. Each of these special cells are to be coloured *red* or *blue*. The edges of the grid (i.e., the topmost row, the bottommost row, the leftmost column, and the rightmost column) will never contain special cells.

Several robots will be placed on cells in the leftmost column and will move according to the colour of their cell:

- If the cell is white, the robot moves right.
- If the cell is red, the robot moves upward.
- If the cell is blue, the robot moves downward.
- If the cell is black, the robot does not move.

Robots do not collide with each other; each robot moves independently of other robots. Multiple robots are allowed to occupy the same cell.

A query consists of two integers a and b ($1 \leq a < b \leq h$). For each $a \leq c \leq b$, there will be a robot starting at $(c, 0)$. Across all possible configurations of colouring the special cells red or blue, determine the minimum number of distinct final cells the robots can occupy.

Note that different queries may result in different configurations of colouring the cells red and blue.

Input Format

Your program must read from standard input.

The first line of input contains two space-separated integers h and w.

The second line of input contains w space-separated integers $x[1], x[2], \ldots, x[w]$, indicating that the special cells are $(x[1], 1), (x[2], 2), \ldots, (x[w], w)$.

The third line of input contains one integer q.

The following q lines of input each contain two space-separated integers. The i-th of these lines contains $a[i]$ and $b[i]$.

Output format

Your program must print to standard output.

The output should contain q lines. The i-th of these lines should contain one integer, the answer to the i-th query.

Subtasks

For all test cases, the input will satisfy the following bounds:

- $1 \leq w, q \leq 200\,000$
- $2 \leq h \leq 200\,000$
- $1 \leq x[j] \leq h$ for all $1 \leq j \leq w$
- $1 \leq a[i] < b[i] \leq h$ for all $1 \leq i \leq q$

Your program will be tested on input instances that satisfy the following restrictions:

Subtask	Marks	Additional Constraints
0	0	Sample test cases
1	10	$h, w \leq 16$, $q \leq 20$
2	4	$a[i] + 1 = b[i]$
3	12	$x[1] < x[2] < \cdots < x[w]$
4	43	$h, w, q \leq 5000$
5	31	No additional constraints

Sample Test Case 1

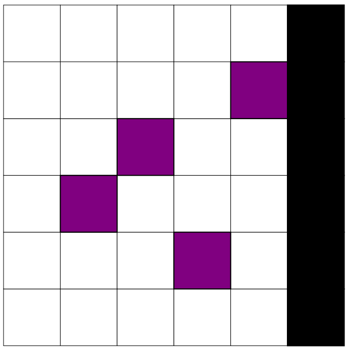
This test case is valid for subtasks 1, 4, and 5.

Input	Output

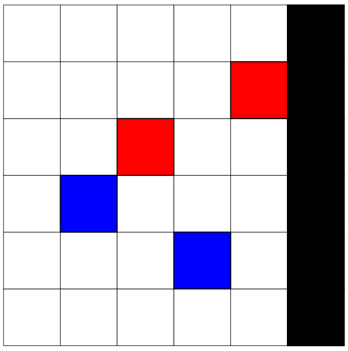
4 4
3 2 4 1
3
1 4
1 3
2 4
	2
	1
	1

Sample Test Case 1 Explanation

The grid is coloured as follows, with the special cells in violet.



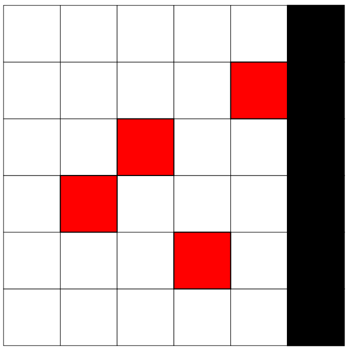
For the first query, we can colour the special cells at $(3, 1)$ and $(4, 3)$ blue, $(2, 2)$ and $(1, 4)$ red to achieve the following:



- The robot starting at $(1,0)$ moves to $(1,1),(1,2),(1,3),(1,4),(0,4),(0,5)$, and ends at $(0,5)$.
- The robot starting at $(2,0)$ moves to $(2,1),(2,2),(1,2),(1,3),(1,4),(0,4),(0,5)$, and ends at $(0,5)$.
- The robot starting at $(3,0)$ moves to $(3,1),(4,1),(4,2),(4,3),(5,3),(5,4)$, and ends at $(5,5)$.
- The robot starting at $(4,0)$ moves to $(4,1),(4,2),(4,3),(5,3),(5,4)$, and ends at $(5,5)$.

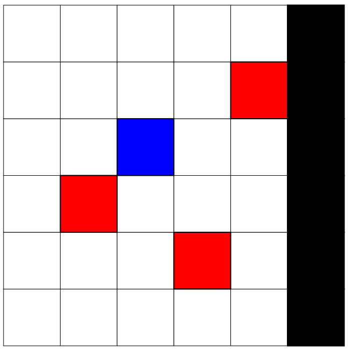
The robots end at 2 distinct cells, $(0,5)$ and $(5,5)$, hence the correct output is 2.

For the second query, we can colour the special cells as follows:



The robots starting at $(1,0)$, $(2,0)$, and $(3,0)$ all end at $(0,5)$.

For the third query, we can colour the special cells as follows:



The robots starting at $(2,0)$, $(3,0)$, and $(4,0)$ all end at $(3,5)$.

Sample Test Case 2

This test case is valid for subtasks 1, 4, and 5.

Input	Output
15 16	2
5 15 3 4 13 8 3 7 14 6 2 10 11 12 9 1	2
20	3
4 10	2
7 15	2
5 15	3
2 6	1
7 15	1
5 15	3
12 15	1
13 14	3
5 14	2
13 14	1
2 11	1
3 11	3
2 5	3
9 11	4
3 15	1
5 14	2
1 13	1
3 7	
7 12	
4 8	