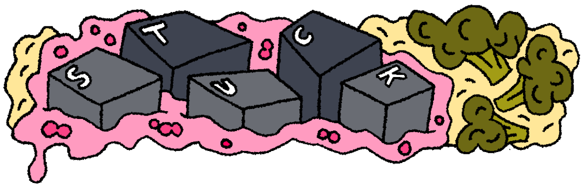
Problem J. JamBrains

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 mebibytes



It's not enough to write the code, you still have to fix three hundred bugs

*Put your cursor at the end of a line in the middle of your code. Press **RUL** arrows. Now repeat it many times. You learned a convenient navigation trick.*

Have you eaten meatballs with cranberry jam? Kostya adores them, especially while solving problems at midnight. In fact, he ate too much in front of his computer: some cranberry jam dripped out of the jar and ruined his keyboard. But he still has to finish solving the problem. And, unlike his keyboard, he doesn't want to be stuck!

The code file that Kostya used consists of n lines numbered from 1 to n, where line i contains s_i possible cursor positions from $(i, 1)$ to (i, s_i), denoting the j-th position on the i-th line as (i, j).

Code navigation with some keys stuck is a bit tricky. Fortunately, Kostya still has Up and Right arrows working:

- Up ($\uparrow$): if the cursor is at position (i, j), then if $i = 1$, the cursor stays in place; otherwise, it moves to position $(i-1, \min(j, s_{i-1}))$, where s_{i-1} is the length of line $i-1$.
- Right ($\rightarrow$): if the cursor is at position (i, j), then if $j < s_i$, the cursor moves to $(i, j+1)$; otherwise, it moves to $(i+1, 1)$; except when at (n, s_n), then it stays in place.

Kostya has a special technique. He picks an initial cursor position and starts to perform an infinite sequence of key presses: u times Up, followed by r times Right, then again u times Up, followed by r times Right, and so on.

For example, let $n = 5$, $u = 2$, $r = 5$, and $s = \{5, 2, 4, 3, 1\}$. Starting from position $(4, 3)$, the first few key presses will move the cursor as follows:

$$(4,3) \uparrow (3,3) \uparrow (2,2) \rightarrow (3,1) \rightarrow (3,2) \rightarrow (3,3) \rightarrow (3,4) \rightarrow (4,1) \ldots$$

A position is called *successful* if, starting from this position and performing his infinite sequence, Kostya will sooner or later have the cursor on the first line.

Your task is to help Kostya count how many successful starting positions exist in his code.

His code isn't finished though, so you also need to handle q updates. In each update, you get two values, pos and x, and you have to assign s_{pos} a new value x. You need to give the answer to the problem before any updates and after processing each update.

Input

The first line contains three integers: n, u, and r ($1 \le n \le 10^5$, $1 \le u, r \le 10^{18}$).

The second line contains n integers $s_1, s_2, \ldots, s_n$ ($1 \le s_i \le 10^{12}$): the number of positions in each line.

The third line contains a single integer q ($1 \le q \le 10^5$), the number of updates.

Each of the next q lines contains two integers, pos and x ($1 \le pos \le n$, $1 \le x \le 10^{12}$): the updates.

The updates are consecutive, so each update is applied to the current state of the code, not the initial one.

Output

Output $q+1$ integers, each on a separate line: the number of successful starting positions before all the updates and after each one of them.

Examples

standard input	standard output
5 2 5 5 2 4 3 1 1 1 1	15 11
5 2 10 5 2 4 3 1 1 2 3	11 12
10 1 42 169 42 42 42 42 42 42 42 42 42 1 2 43	211 254

Note

In the first example, before any updates, starting from position $(5,1)$ and pressing "2 times Up followed by 5 times Right" in a loop:

Round 1: $(5,1) \uparrow (4,1) \uparrow (3,1) \rightarrow (3,2) \rightarrow (3,3) \rightarrow (3,4) \rightarrow (4,1) \rightarrow (4,2)$

Round 2: $(4,2) \uparrow (3,2) \uparrow (2,2) \rightarrow (3,1) \rightarrow (3,2) \rightarrow (3,3) \rightarrow (3,4) \rightarrow (4,1)$

Round 3: $(4,1) \uparrow (3,1) \uparrow (2,1) \rightarrow (2,2) \rightarrow (3,1) \rightarrow (3,2) \rightarrow (3,3) \rightarrow (3,4)$

Round 4: $(3,4) \uparrow (2,2) \uparrow (\mathbf{1},\mathbf{2}) \rightarrow \dots$

Thus, $(5,1)$ is a successful position.

On the image below you can see the starting positions of each round marked with the green dots and the final position marked with the green star.