

Codeforces Round #248 (Div. 2)**A. Kitahara Haruki's Gift**

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Kitahara Haruki has bought n apples for Touma Kazusa and Ogiso Setsuna. Now he wants to divide all the apples between the friends.

Each apple weights 100 grams or 200 grams. Of course Kitahara Haruki doesn't want to offend any of his friend. Therefore the total weight of the apples given to Touma Kazusa must be equal to the total weight of the apples given to Ogiso Setsuna.

But unfortunately Kitahara Haruki doesn't have a knife right now, so he cannot split any apple into some parts. Please, tell him: is it possible to divide all the apples in a fair way between his friends?

Input

The first line contains an integer n ($1 \leq n \leq 100$) — the number of apples. The second line contains n integers w_1, w_2, \dots, w_n ($w_i = 100$ or $w_i = 200$), where w_i is the weight of the i -th apple.

Output

In a single line print "YES" (without the quotes) if it is possible to divide all the apples between his friends. Otherwise print "NO" (without the quotes).

Sample test(s)

| |
|------------------|
| input |
| 3 100 200 100 |
| output |
| YES |

| |
|----------------------|
| input |
| 4 100 100 100 200 |
| output |
| NO |

Note

In the first test sample Kitahara Haruki can give the first and the last apple to Ogiso Setsuna and the middle apple to Touma Kazusa.

B. Kuriyama Mirai's Stones

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Kuriyama Mirai has killed many monsters and got many (namely n) stones. She numbers the stones from 1 to n . The cost of the i -th stone is v_i . Kuriyama Mirai wants to know something about these stones so she will ask you two kinds of questions:

1. She will tell you two numbers, l and r ($1 \leq l \leq r \leq n$), and you should tell her $\sum_{i=l}^r v_i$.
2. Let u_i be the cost of the i -th cheapest stone (the cost that will be on the i -th place if we arrange all the stone costs in non-decreasing order). This time she will tell you two numbers, l and r ($1 \leq l \leq r \leq n$), and you should tell her $\sum_{i=l}^r u_i$.

For every question you should give the correct answer, or Kuriyama Mirai will say "fuyukai desu" and then become unhappy.

Input

The first line contains an integer n ($1 \leq n \leq 10^5$). The second line contains n integers: v_1, v_2, \dots, v_n ($1 \leq v_i \leq 10^9$) — costs of the stones.

The third line contains an integer m ($1 \leq m \leq 10^5$) — the number of Kuriyama Mirai's questions. Then follow m lines, each line contains three integers $type, l$ and r ($1 \leq l \leq r \leq n$; $1 \leq type \leq 2$), describing a question. If $type$ equal to 1, then you should output the answer for the first question, else you should output the answer for the second one.

Output

Print m lines. Each line must contain an integer — the answer to Kuriyama Mirai's question. Print the answers to the questions in the order of input.

Sample test(s)

| |
|--|
| input |
| 6 6 4 2 7 2 7 3 2 3 6 1 3 4 1 1 6 |
| output |
| 24 9 28 |

| |
|--|
| input |
| 4 5 5 2 3 10 1 2 4 2 1 4 1 1 1 2 1 4 2 1 2 1 1 1 1 3 3 1 1 3 1 4 4 1 2 2 |
| output |
| 10 15 5 15 5 5 2 12 3 5 |

Note

Please note that the answers to the questions may overflow 32-bit integer type.

C. Ryouko's Memory Note

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Ryouko is an extremely forgetful girl, she could even forget something that has just happened. So in order to remember, she takes a notebook with her, called *Ryouko's Memory Note*. She writes what she sees and what she hears on the notebook, and the notebook became her memory.

Though Ryouko is forgetful, she is also born with superb analyzing abilities. However, analyzing depends greatly on gathered information, in other words, memory. So she has to shuffle through her notebook whenever she needs to analyze, which is tough work.

Ryouko's notebook consists of n pages, numbered from 1 to n . To make life (and this problem) easier, we consider that to turn from page x to page y , $|x - y|$ pages should be turned. During analyzing, Ryouko needs m pieces of information, the i -th piece of information is on page a_i . Information must be read from the notebook in order, so the total number of pages that Ryouko needs to turn is $\sum_{i=1}^{m-1} |a_{i+1} - a_i|$.

Ryouko wants to decrease the number of pages that need to be turned. In order to achieve this, she can merge two pages of her notebook. If Ryouko merges page x to page y , she would copy all the information on page x to y ($1 \leq x, y \leq n$), and consequently, all elements in sequence a that was x would become y . Note that x can be equal to y , in which case no changes take place.

Please tell Ryouko the minimum number of pages that she needs to turn. Note she can apply the described operation at most once before the reading. Note that the answer can exceed 32-bit integers.

Input

The first line of input contains two integers n and m ($1 \leq n, m \leq 10^5$).

The next line contains m integers separated by spaces: a_1, a_2, \dots, a_m ($1 \leq a_i \leq n$).

Output

Print a single integer — the minimum number of pages Ryouko needs to turn.

Sample test(s)

| |
|--------------------|
| input |
| 4 6 1 2 3 4 3 2 |
| output |
| 3 |
| input |
| 10 5 9 4 3 8 8 |
| output |
| 6 |

Note

In the first sample, the optimal solution is to merge page 4 to 3, after merging sequence a becomes $\{1, 2, 3, 3, 3, 2\}$, so the number of pages Ryouko needs to turn is $|1 - 2| + |2 - 3| + |3 - 3| + |3 - 3| + |3 - 2| = 3$.

In the second sample, optimal solution is achieved by merging page 9 to 4.

D. Nanami's Digital Board

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Nanami is an expert at playing games. This day, Nanami's good friend Hajime invited her to watch a game of baseball. Unwilling as she was, she followed him to the stadium. But Nanami had no interest in the game, so she looked around to see if there was something that might interest her. That's when she saw the digital board at one end of the stadium.

The digital board is n pixels in height and m pixels in width, every pixel is either light or dark. The pixels are described by its coordinate. The j -th pixel of the i -th line is pixel (i, j) . The board displays messages by switching a combination of pixels to light, and the rest to dark. Nanami notices that the state of the pixels on the board changes from time to time. At certain times, certain pixels on the board may switch from light to dark, or from dark to light.

Nanami wonders, what is the area of the biggest light block such that a specific pixel is on its side. A light block is a sub-rectangle of the board, in which all pixels are light. Pixel (i, j) belongs to a side of sub-rectangle with (x_1, y_1) and (x_2, y_2) as its upper-left and lower-right vertex if and only if it satisfies the logical condition:

$$((i = x_1 \text{ or } i = x_2) \text{ and } (y_1 \leq j \leq y_2)) \text{ or } ((j = y_1 \text{ or } j = y_2) \text{ and } (x_1 \leq i \leq x_2)).$$

Nanami has all the history of changing pixels, also she has some questions of the described type, can you answer them?

Input

The first line contains three space-separated integers n , m and q ($1 \leq n, m, q \leq 1000$) — the height and width of the digital board, and the number of operations.

Then follow n lines, each line containing m space-separated integers. The j -th integer of the i -th line is $a_{i,j}$ — the initial state of pixel (i, j) .

- If $a_{i,j} = 0$, pixel (i, j) is initially dark.
- If $a_{i,j} = 1$, pixel (i, j) is initially light.

Then follow q lines, each line containing three space-separated integers op , x , and y ($1 \leq op \leq 2$; $1 \leq x \leq n$; $1 \leq y \leq m$), describing an operation.

- If $op = 1$, the pixel at (x, y) changes its state (from light to dark or from dark to light).
- If $op = 2$, Nanami queries the biggest light block with pixel (x, y) on its side.

Output

For each query, print a single line containing one integer — the answer to Nanami's query.

Sample test(s)

| |
|---|
| input |
| 3 4 5 0 1 1 0 1 0 0 1 0 1 1 0 2 2 2 2 1 2 1 2 2 1 2 3 2 2 2 |
| output |
| 0 2 6 |

| |
|--|
| input |
| 3 3 4 1 1 1 1 1 1 1 1 1 2 2 2 1 2 2 2 1 1 2 2 1 |
| output |
| 6 3 3 |

Note

Consider the first sample.

The first query specifies pixel $(2, 2)$, which is dark itself, so there are no valid light blocks, thus the answer is 0.

The second query specifies pixel $(1, 2)$. The biggest light block is the block with $(1, 2)$ as its upper-left vertex and $(1, 3)$ as its lower-right vertex.

The last query specifies pixel $(2, 2)$, which became light in the third operation. The biggest light block is the block with $(1, 2)$ as its upper-left vertex and $(3, 3)$ as its lower-right vertex.

E. Tachibana Kanade's Tofu

time limit per test: 5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Tachibana Kanade likes Mapo Tofu very much. One day, the canteen cooked all kinds of tofu to sell, but not all tofu is Mapo Tofu, only those spicy enough can be called Mapo Tofu.

Each piece of tofu in the canteen is given a m -based number, all numbers are in the range $[l, r]$ (l and r being m -based numbers), and for every m -based integer in the range $[l, r]$, there exists a piece of tofu with that number.

To judge what tofu is Mapo Tofu, Tachibana Kanade chose n m -based number strings, and assigned a value to each string. If a string appears in the number of a tofu, the value of the string will be added to the value of that tofu. If a string appears multiple times, then the value is also added that many times. Initially the value of each tofu is zero.

Tachibana Kanade considers tofu with values no more than k to be Mapo Tofu. So now Tachibana Kanade wants to know, how many pieces of tofu are Mapo Tofu?

Input

The first line contains three integers n , m and k ($1 \leq n \leq 200$; $2 \leq m \leq 20$; $1 \leq k \leq 500$). Where n denotes the number of strings, m denotes the base used, and k denotes the limit of the value for Mapo Tofu.

The second line represents the number l . The first integer in the line is len ($1 \leq len \leq 200$), describing the length (number of digits in base m) of l . Then follow len integers a_1, a_2, \dots, a_{len} ($0 \leq a_i < m$; $a_1 > 0$) separated by spaces, representing the digits of l , with a_1 being the highest digit and a_{len} being the lowest digit.

The third line represents the number r in the same format as l . It is guaranteed that $1 \leq l \leq r$.

Then follow n lines, each line describing a number string. The i -th line contains the i -th number string and v_i — the value of the i -th string ($1 \leq v_i \leq 200$). All number strings are described in almost the same format as l , the only difference is number strings may contain necessary leading zeros (see the first example). The sum of the lengths of all number strings does not exceed 200.

Output

Output the number of pieces of Mapo Tofu modulo 1000000007 ($10^9 + 7$). The answer should be a decimal integer.

Sample test(s)

| |
|--|
| input |
| 2 10 1 1 1 3 1 0 0 1 1 1 1 0 1 |
| output |
| 97 |

| |
|---|
| input |
| 2 10 12 2 5 9 6 6 3 5 4 9 7 2 0 6 1 3 6 7 2 1 |
| output |
| 635439 |

| |
|---|
| input |
| 4 2 6 6 1 0 1 1 1 0 6 1 1 0 1 0 0 1 1 2 3 0 1 0 5 4 0 1 1 0 4 3 1 0 1 2 |
| output |
| 2 |

Note

In the first sample, 10, 11 and 100 are the only three decimal numbers in $[1, 100]$ with a value greater than 1. Here the value of 1 is 1 but not 2, since numbers cannot contain leading zeros and thus cannot be written as "01".

In the second sample, no numbers in the given interval have a value greater than 12.

In the third sample, 110000 and 110001 are the only two binary numbers in the given interval with a value no greater than 6.

