A. 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 $(1 \le x, y \le n)$, 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 $(1 \le n, m \le 10^5)$.

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

Output

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

Examples

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.

B. 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:

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

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 $(1 \le n, m, q \le 1000)$ — 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 $(1 \le op \le 2;\ 1 \le x \le n;\ 1 \le y \le m)$, 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.

Examples

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.

C. 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 $(1 \le n \le 200;\ 2 \le m \le 20;\ 1 \le k \le 500)$. 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 $(1 \le len \le 200)$, describing the length (number of digits in base m) of l. Then follow len integers $a_1, a_2, \ldots, a_{len}$ $(0 \le a_i < m;\ a_1 > 0)$ 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 $1 \le l \le r$.

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 $(1 \le v_i \le 200)$. 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.

Examples

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.

D. Nanami's Power Plant

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

Nanami likes playing games, and is also really good at it. This day she was playing a new game which involved operating a power plant. Nanami's job is to control the generators in the plant and produce maximum output.

There are n generators in the plant. Each generator should be set to a generating level. Generating level is an integer (possibly zero or negative), the generating level of the i-th generator should be between l_i and r_i (both inclusive). The output of a generator can be calculated using a certain quadratic function $f(x)$, where x is the generating level of the generator. Each generator has its own function, the function of the i-th generator is denoted as $f_i(x)$.

However, there are m further restrictions to the generators. Let the generating level of the i-th generator be x_i. Each restriction is of the form $x_u \le x_v + d$, where u and v are IDs of two different generators and d is an integer.

Nanami found the game tedious but giving up is against her creed. So she decided to have a program written to calculate the answer for her (the maximum total output of generators). Somehow, this became your job.

Input

The first line contains two integers n and m $(1 \le n \le 50;\ 0 \le m \le 100)$ — the number of generators and the number of restrictions.

Then follow n lines, each line contains three integers a_i, b_i, and c_i $(|a_i| \le 10;\ |b_i|, |c_i| \le 1000)$ — the coefficients of the function $f_i(x)$. That is, $f_i(x) = a_i x^2 + b_i x + c_i$.

Then follow another n lines, each line contains two integers l_i and r_i $(-100 \le l_i \le r_i \le 100)$.

Then follow m lines, each line contains three integers u_i, v_i, and d_i $(1 \le u_i, v_i \le n;\ u_i \ne v_i;\ |d_i| \le 200)$, describing a restriction. The i-th restriction is $x_{u_i} \le x_{v_i} + d_i$.

Output

Print a single line containing a single integer — the maximum output of all the generators. It is guaranteed that there exists at least one valid configuration.

Examples

input

```
3 3
0 1 0
0 1 1
0 1 2
0 3
1 2
-100 100
1 2 0
2 3 0
3 1 0
```

output

```
9
```

input

```
5 8
1 -8 20
2 -4 0
-1 10 -10
0 1 0
0 -1 1
1 9
1 4
0 10
3 11
7 9
2 1 3
1 2 3
2 3 3
3 2 3
3 4 3
4 3 3
4 5 3
5 4 3
```

output

```
46
```

Note

In the first sample, $f_1(x) = x$, $f_2(x) = x + 1$, and $f_3(x) = x + 2$, so we are to maximize the sum of the generating levels. The restrictions are $x_1 \le x_2$, $x_2 \le x_3$, and $x_3 \le x_1$, which gives us $x_1 = x_2 = x_3$. The optimal configuration is $x_1 = x_2 = x_3 = 2$, which produces an output of 9.

In the second sample, restrictions are equal to $|x_i - x_{i+1}| \le 3$ for $1 \le i < n$. One of the optimal configurations is $x_1 = 1$, $x_2 = 4$, $x_3 = 5$, $x_4 = 8$ and $x_5 = 7$.

E. Furukawa Nagisa's Tree

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

One day, Okazaki Tomoya has bought a tree for Furukawa Nagisa's birthday. The tree is so strange that every node of the tree has a value. The value of the i-th node is v_i. Now Furukawa Nagisa and Okazaki Tomoya want to play a game on the tree.

Let (s, e) be the path from node s to node e, we can write down the sequence of the values of nodes on path (s, e), and denote this sequence as $S(s, e)$. We define the value of the sequence $G(S(s, e))$ as follows. Suppose that the sequence is $z_0, z_1...z_{l-1}$, where l is the length of the sequence. We define $G(S(s, e)) = z_0 \times k^0 + z_1 \times k^1 + ... + z_{l-1} \times k^{l-1}$. If the path (s, e) satisfies $G(S(s,e)) \equiv x (\mathrm{mod}\ y)$, then the path (s, e) belongs to Furukawa Nagisa, otherwise it belongs to Okazaki Tomoya.

Calculating who has more paths is too easy, so they want to play something more difficult. Furukawa Nagisa thinks that if paths (p_1, p_2) and (p_2, p_3) belong to her, then path (p_1, p_3) belongs to her as well. Also, she thinks that if paths (p_1, p_2) and (p_2, p_3) belong to Okazaki Tomoya, then path (p_1, p_3) belongs to Okazaki Tomoya as well. But in fact, this conclusion isn't always right. So now Furukawa Nagisa wants to know how many triplets (p_1, p_2, p_3) are correct for the conclusion, and this is your task.

Input

The first line contains four integers n, y, k and x $(1 \le n \le 10^5; 2 \le y \le 10^9; 1 \le k < y; 0 \le x < y)$ — n being the number of nodes on the tree. It is guaranteed that y is a prime number.

The second line contains n integers, the i-th integer is v_i $(0 \le v_i < y)$.

Then follow $n - 1$ lines, each line contains two integers, denoting an edge of the tree. The nodes of the tree are numbered from 1 to n.

Output

Output a single integer — the number of triplets that are correct for Furukawa Nagisa's conclusion.

Examples

input
```
1 2 1 0
1
```
output
```
1
```

input
```
3 5 2 1
4 3 1
1 2
2 3
```
output
```
14
```

input
```
8 13 8 12
0 12 7 4 12 0 8 12
1 8
8 4
4 6
6 2
2 3
8 5
2 7
```
output
```
341
```

Codeforces (c) Copyright 2010-2016 Mike Mirzayanov
The only programming contests Web 2.0 platform