A. Inna and Alarm Clock

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Inna loves sleeping very much, so she needs n alarm clocks in total to wake up. Let's suppose that Inna's room is a 100×100 square with the lower left corner at point $(0, 0)$ and with the upper right corner at point $(100, 100)$. Then the alarm clocks are points with integer coordinates in this square.

The morning has come. All n alarm clocks in Inna's room are ringing, so Inna wants to turn them off. For that Inna has come up with an amusing game:

- First Inna chooses a type of segments that she will use throughout the game. The segments can be either vertical or horizontal.
- Then Inna makes multiple moves. In a single move, Inna can paint a segment of any length on the plane, she chooses its type at the beginning of the game (either vertical or horizontal), then all alarm clocks that are on this segment switch off. The game ends when all the alarm clocks are switched off.

Inna is very sleepy, so she wants to get through the alarm clocks as soon as possible. Help her, find the minimum number of moves in the game that she needs to turn off all the alarm clocks!

Input

The first line of the input contains integer n $(1 \le n \le 10^5)$ — the number of the alarm clocks. The next n lines describe the clocks: the i-th line contains two integers x_i, y_i — the coordinates of the i-th alarm clock $(0 \le x_i, y_i \le 100)$.

Note that a single point in the room can contain any number of alarm clocks and the alarm clocks can lie on the sides of the square that represents the room.

Output

In a single line print a single integer — the minimum number of segments Inna will have to draw if she acts optimally.

Examples

input

```
4
0 0
0 1
0 2
1 0
```

output

```
2
```

input

```
4
0 0
0 1
1 0
1 1
```

output

```
2
```

input

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

output

```
3
```

Note

In the first sample, Inna first chooses type "vertical segments", and then she makes segments with ends at : $(0, 0)$, $(0, 2)$; and, for example, $(1, 0)$, $(1, 1)$. If she paints horizontal segments, she will need at least 3 segments.

In the third sample it is important to note that Inna doesn't have the right to change the type of the segments during the game. That's why she will need 3 horizontal or 3 vertical segments to end the game.

B. Inna, Dima and Song

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Inna is a great piano player and Dima is a modest guitar player. Dima has recently written a song and they want to play it together. Of course, Sereja wants to listen to the song very much.

A song is a sequence of notes. Dima and Inna want to play each note at the same time. At that, they can play the i-th note at volume v ($1 \le v \le a_i$; v is an integer) both on the piano and the guitar. They should retain harmony, so the total volume with which the i-th note was played on the guitar and the piano must equal b_i. If Dima and Inna cannot play a note by the described rules, they skip it and Sereja's joy drops by 1. But if Inna and Dima play the i-th note at volumes x_i and y_i $(x_i + y_i = b_i)$ correspondingly, Sereja's joy rises by $x_i \cdot y_i$.

Sereja has just returned home from the university and his current joy is 0. Help Dima and Inna play the song so as to maximize Sereja's total joy after listening to the whole song!

Input

The first line of the input contains integer n $(1 \le n \le 10^5)$ — the number of notes in the song. The second line contains n integers a_i $(1 \le a_i \le 10^6)$. The third line contains n integers b_i $(1 \le b_i \le 10^6)$.

Output

In a single line print an integer — the maximum possible joy Sereja feels after he listens to a song.

Examples

input

```
3
1 1 2
2 2 3
```

output

```
4
```

input

```
1
2
5
```

output

```
-1
```

Note

In the first sample, Dima and Inna play the first two notes at volume 1 ($1 + 1 = 2$, the condition holds), they should play the last note at volumes 1 and 2. Sereja's total joy equals: $1 \cdot 1 + 1 \cdot 1 + 1 \cdot 2 = 4$.

In the second sample, there is no such pair (x, y), that $1 \le x, y \le 2$, $x + y = 5$, so Dima and Inna skip a note. Sereja's total joy equals -1.

C. Inna and Candy Boxes

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Inna loves sweets very much. She has n closed present boxes lines up in a row in front of her. Each of these boxes contains either a candy (Dima's work) or nothing (Sereja's work). Let's assume that the boxes are numbered from 1 to n, from left to right.

As the boxes are closed, Inna doesn't know which boxes contain candies and which boxes contain nothing. Inna chose number k and asked w questions to Dima to find that out. Each question is characterised by two integers l_i, r_i ($1 \le l_i \le r_i \le n$; $r - l + 1$ is divisible by k), the i-th question is: "Dima, is that true that among the boxes with numbers from l_i to r_i, inclusive, the candies lie **only** in boxes with numbers $l_i + k - 1$, $l_i + 2k - 1$, $l_i + 3k - 1$, ..., r_i?"

Dima hates to say "no" to Inna. That's why he wonders, what number of actions he will have to make for each question to make the answer to the question positive. In one action, Dima can either secretly take the candy from any box or put a candy to any box (Dima has infinitely many candies). Help Dima count the number of actions for each Inna's question.

Please note that Dima doesn't change the array during Inna's questions. That's why when you calculate the number of operations for the current question, please assume that the sequence of boxes didn't change.

Input

The first line of the input contains three integers n, k and w ($1 \le k \le min(n, 10), 1 \le n, w \le 10^5$). The second line contains n characters. If the i-th box contains a candy, the i-th character of the line equals 1, otherwise it equals 0.

Each of the following w lines contains two integers l_i and r_i ($1 \le l_i \le r_i \le n$) — the description of the i-th question. It is guaranteed that $r_i - l_i + 1$ is divisible by k.

Output

For each question, print a single number on a single line — the minimum number of operations Dima needs to make the answer to the question positive.

Examples

input

```
10 3 3
1010100011
1 3
1 6
4 9
```

output

```
1
3
2
```

Note

For the first question, you need to take a candy from the first box to make the answer positive. So the answer is 1.

For the second question, you need to take a candy from the first box, take a candy from the fifth box and put a candy to the sixth box. The answer is 3.

For the third question, you need to take a candy from the fifth box and put it to the sixth box. The answer is 2.

D. Inna and Sweet Matrix

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Inna loves sweets very much. That's why she decided to play a game called "Sweet Matrix".

Inna sees an $n \times m$ matrix and k candies. We'll index the matrix rows from 1 to n and the matrix columns from 1 to m. We'll represent the cell in the i-th row and j-th column as (i, j). Two cells (i, j) and (p, q) of the matrix are *adjacent* if $|i - p| + |j - q| = 1$. A *path* is a sequence of the matrix cells where each pair of neighbouring cells in the sequence is adjacent. We'll call the number of cells in the sequence the path's length.

Each cell of the matrix can have at most one candy. Initiallly, all the cells are empty. Inna is trying to place each of the k candies in the matrix one by one. For each candy Inna chooses cell (i, j) that will contains the candy, and also chooses the path that starts in cell $(1, 1)$ and ends in cell (i, j) and doesn't contain any candies. After that Inna moves the candy along the path from cell $(1, 1)$ to cell (i, j), where the candy stays forever. If at some moment Inna can't choose a path for the candy, she loses. If Inna can place all the candies in the matrix in the described manner, then her penalty equals the sum of lengths of all the paths she has used.

Help Inna to minimize the penalty in the game.

Input

The first line of the input contains three integers n, m and k $(1 \le n, m \le 50, 1 \le k \le n \cdot m)$.

Output

In the first line print an integer — Inna's minimum penalty in the game.

In the next k lines print the description of the path for each candy. The description of the path of the candy that is placed i-th should follow on the i-th line. The description of a path is a sequence of cells. Each cell must be written in the format (i, j), where i is the number of the row and j is the number of the column. You are allowed to print extra whitespaces in the line. If there are multiple optimal solutions, print any of them.

Please follow the output format strictly! If your program passes the first pretest, then the output format is correct.

Examples

input

```
4 4 4
```

output

```
8
(1,1) (2,1) (2,2)
(1,1) (1,2)
(1,1) (2,1)
(1,1)
```

Note

Note to the sample. Initially the matrix is empty. Then Inna follows her first path, the path penalty equals the number of cells in it — 3. Note that now no path can go through cell (2, 2), as it now contains a candy. The next two candies go to cells (1, 2) and (2, 1). Inna simply leaves the last candy at cell (1, 1), the path contains only this cell. The total penalty is: 3 + 2 + 2 + 1 = 8.

Note that Inna couldn't use cell (1, 1) to place, for instance, the third candy as in this case she couldn't have made the path for the fourth candy.

E. Inna and Large Sweet Matrix

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

Inna loves sweets very much. That's why she wants to play the "Sweet Matrix" game with Dima and Sereja. But Sereja is a large person, so the game proved small for him. Sereja suggested playing the "Large Sweet Matrix" game.

The "Large Sweet Matrix" playing field is an $n \times m$ matrix. Let's number the rows of the matrix from 1 to n, and the columns — from 1 to m. Let's denote the cell in the i-th row and j-th column as (i, j). Each cell of the matrix can contain multiple candies, initially all cells are empty. The game goes in w moves, during each move one of the two following events occurs:

1. Sereja chooses five integers x_1, y_1, x_2, y_2, v $(x_1 \le x_2, y_1 \le y_2)$ and adds v candies to each matrix cell (i, j) $(x_1 \le i \le x_2;\ y_1 \le j \le y_2)$.
2. Sereja chooses four integers x_1, y_1, x_2, y_2 $(x_1 \le x_2, y_1 \le y_2)$. Then he asks Dima to calculate the total number of candies in cells (i, j) $(x_1 \le i \le x_2;\ y_1 \le j \le y_2)$ and he asks Inna to calculate the total number of candies in the cells of matrix (p, q), which meet the following logical criteria: ($p < x_1$ OR $p > x_2$) AND ($q < y_1$ OR $q > y_2$). Finally, Sereja asks to write down the difference between the number Dima has calculated and the number Inna has calculated (D - I).

Unfortunately, Sereja's matrix is really huge. That's why Inna and Dima aren't coping with the calculating. Help them!

Input

The first line of the input contains three integers n, m and w $(3 \le n, m \le 4 \cdot 10^6;\ 1 \le w \le 10^5)$.

The next w lines describe the moves that were made in the game.

- A line that describes an event of the first type contains 6 integers: 0, x_1, y_1, x_2, y_2 and v $(1 \le x_1 \le x_2 \le n;\ 1 \le y_1 \le y_2 \le m;\ 1 \le v \le 10^9)$.
- A line that describes an event of the second type contains 5 integers: 1, x_1, y_1, x_2, y_2 $(2 \le x_1 \le x_2 \le n - 1;\ 2 \le y_1 \le y_2 \le m - 1)$.

It is guaranteed that the second type move occurs at least once. It is guaranteed that a single operation will not add more than 10^9 candies.

Be careful, the constraints are very large, so please use optimal data structures. Max-tests will be in pretests.

Output

For each second type move print a single integer on a single line — the difference between Dima and Inna's numbers.

Examples

input

```
4 5 5
0 1 1 2 3 2
0 2 2 3 3 3
0 1 5 4 5 1
1 2 3 3 4
1 3 4 3 4
```

output

```
2
-21
```

Note

Note to the sample. After the first query the matrix looks as:

```
22200
22200
00000
00000
```

After the second one it is:

```
22200
25500
03300
00000
```

After the third one it is:

```
22201
25501
03301
00001
```

For the fourth query, Dima's sum equals 5 + 0 + 3 + 0 = 8 and Inna's sum equals 4 + 1 + 0 + 1 = 6. The answer to the query equals 8 - 6 = 2. For the fifth query, Dima's sum equals 0 and Inna's sum equals 18 + 2 + 0 + 1 = 21. The answer to the query is 0 - 21 = -21.