

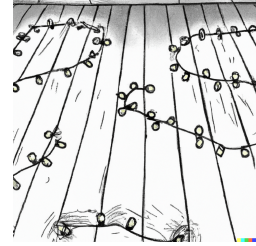


Task Lampice

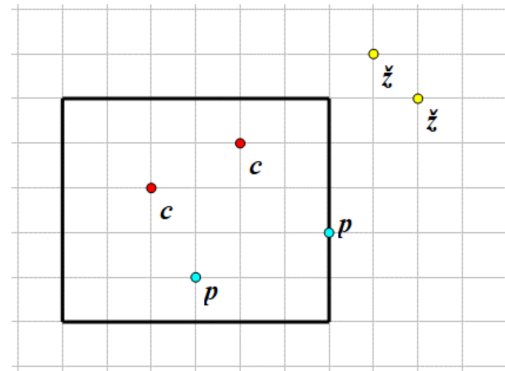
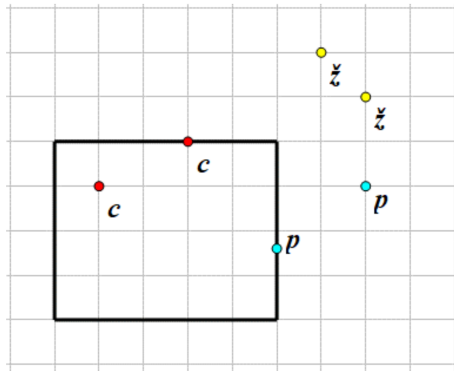
Christmas is coming! Teo has already decided to decorate his terrace.

Teo has a big rectangular-shaped terrace. It is n meters long and m meters wide. Teo has decided to decorate his terrace in a very strange way. Instead of hanging Christmas lights on the edges of his terrace, he will put them on the floor!

Teo has $2k$ lamps, two per each of k colours. He will put each in some position (x_i, y_i) , where x_i represents the distance from the left side of the terrace and y_i from the bottom side.



Proud of how he decorated the terrace, he decided to take the rest of his day off. But soon he got bored, so he returned to the terrace. He started counting nice rectangles on the terrace. A rectangle is nice if for each colour both lamps are either inside or outside of the rectangle. If a lamp is located on the rectangle edge, it is considered to be inside of it.



*The left rectangle is **not** nice. One blue lamp is inside the rectangle, and one is outside.
The right rectangle is nice. Red and blue lamps are inside. Yellow lamps are outside.*

Teo has realized that counting nice rectangles is not an easy job. He is interested in how many nice rectangles are there, whose corners have integer distances from the bottom and left sides of the terrace. All rectangles we consider are **parallel** with terrace sides. This is where you step in! Count the number of nice rectangles.

Input

The first line contains three integers n, m, k ($1 \leq n \leq 150, 1 \leq m \leq 1\,000, 0 \leq k \leq 200\,000$), the length and the width of the terrace, and the number of lamp colours.

The next k lines contain four numbers x_1, y_1, x_2, y_2 ($0 \leq x_1, x_2 \leq n, 0 \leq y_1, y_2 \leq m$), positions of the first and the second lamp of the i -th colour.

Output

In a single line output the number of nice rectangles.



Scoring

Subtask	Points	Constraints
1	26	$x_1 = y_1 = 0$ for each lamp colour
2	12	$n, m \leq 10, k \leq 1\,000$
3	35	$m \leq 150$
4	37	No additional constraints.

Examples

input

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

output

```
3
```

input

```
3 3 0
```

output

```
36
```

input

```
3 3 5
0 0 0 0
0 0 1 3
0 0 3 1
1 3 3 1
1 3 3 1
```

output

```
7
```

Clarification of the first example:

The image shows all nice rectangles from the first example.

