Collision Free Swarming

Input File	Output File	Time Limit	Memory Limit
standard input	standard output	3 seconds	256 MiB

Your local university recently purchased N microbots to simulate and study swarm behaviour. The microbots are placed on a testing field which can be thought of as a grid of squares with R rows (numbered from 1 to R from top to bottom) and C columns (numbered from 1 to C from left to right). The i-th microbot begins in the square in the r_i-th row and c_i-th column. No two microbots begin in the same square.

The researchers will choose a *swarm square* for the microbots to swarm to. Once the square is chosen, each microbot will begin moving from their initial square to the swarm square at the rate of one square per second.

The microbots move simultaneously each second according to the following rules:

- If a microbot is already at the swarm square, it does nothing.
- Otherwise, the microbot chooses one of the four adjacent squares (up, down, left, right) to move to.
- Each microbot always moves so as to minimize the time it takes to reach the swarm square. If there are multiple moves that minimize the time, it chooses randomly.

If two robots are ever at the same square at the same time (except at the swarm square), then they will *collide*, breaking the robots! A swarm square is *good* if no two microbots collide, no matter how they choose to move.

Some examples are shown in Figure 1 below. The examples in the top-left and top-right show good swarm squares, while the examples in the bottom-left and bottom-right show swarm squares that are not good. How many good swarm squares are there?

Subtasks and Constraints

For all subtasks, you are guaranteed that:

- $2 \leq N \leq 1000$.
- $2 \leq R, C \leq 100\,000$.
- $1 \leq r_i \leq R$, for all i.
- $1 \leq c_i \leq C$, for all i.
- No two microbots begin on the same square.

Additional constraints for each subtask are given below.

Subtask	Points	Additional constraints
1	5	$C \leq 1000$ and $r_i = 1$, for all i.
2	5	$r_i = 1$, for all i.
3	10	$r_i = c_i$, for all i.
4	20	$N = 2$.
5	15	$R, C, N \leq 30$.
6	25	$R, C, N \leq 300$.
7	15	$R, C \leq 2000$.
8	5	No special constraints.

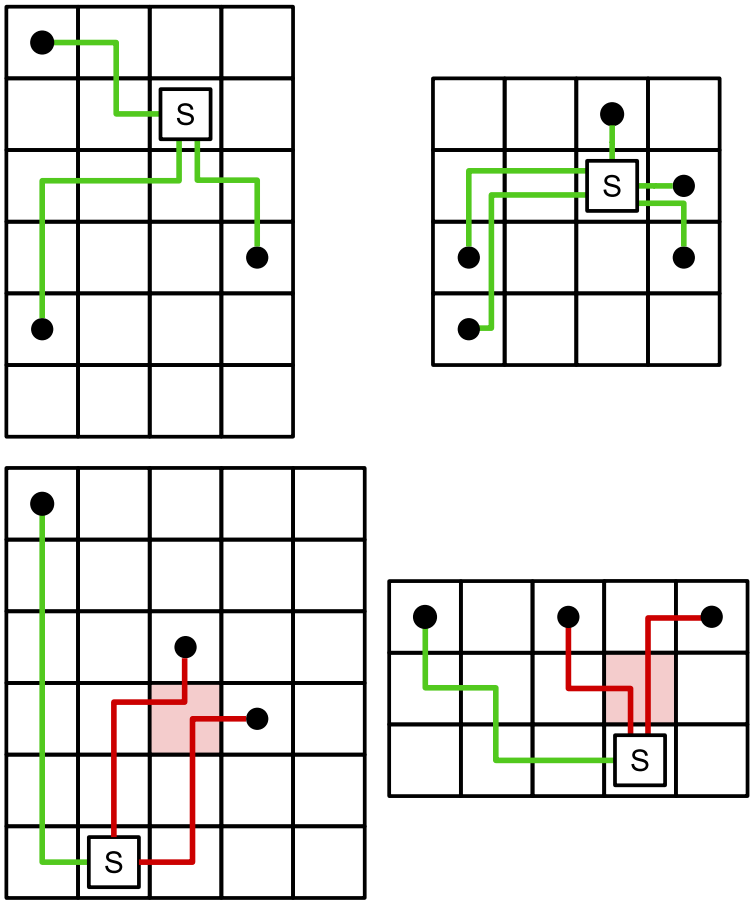

Figure 1: Black circles represent microbots and the square marked ‘S’ represents the chosen swarm square.

Input

- The first line of input contains the three integers, N, R and C.
- Then N lines follow. The i-th line contains the two integers r_i and c_i.

Output

Output a single integer, the number of good swarm squares.

Note: The answer can be quite large. Consider using 64-bit integers (a `long long` in C++).

Sample Input 1

```
3 6 4
4 4
1 1
5 1
```

Sample Input 2

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

Sample Input 3

```
3 6 5
4 4
3 3
1 1
```

Sample Input 4

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

Sample Output 1

```
16
```

Sample Output 2

```
5
```

Sample Output 3

```
17
```

Sample Output 4

```
9
```

Explanation

Sample cases 1, 2, 3 and 4 correspond to the examples in the top-left, top-right, bottom-left and bottom-right of Figure 1 above.

Note that sample case 3 is an example of a possible case in subtask 3, and sample case 4 is an example of a possible case in subtask 2.

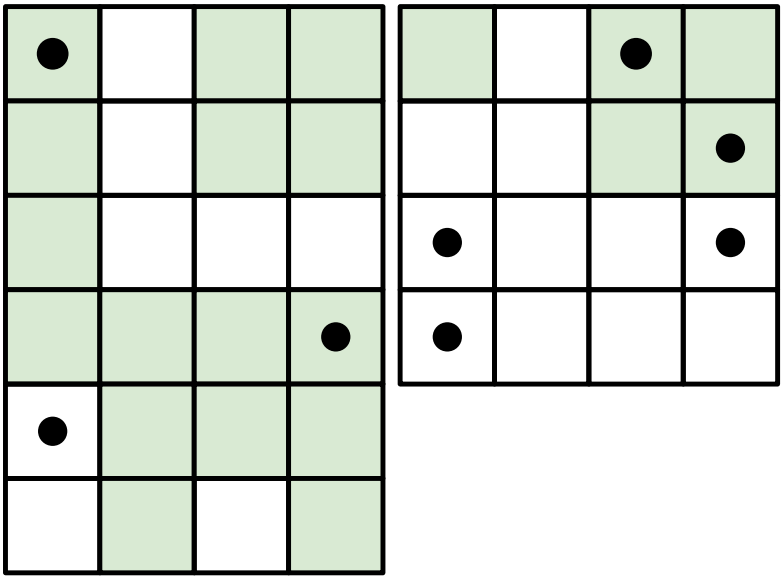

Figure 2: The good swarm squares in sample case 1 and sample case 2 are shown above. Sample cases 3 and 4 are left as an exercise to the reader.