

The city of Gridland is represented as an $n \times m$ matrix where the rows are numbered from 1 to n and the columns are numbered from 1 to m .

Gridland has a network of train tracks that always run in straight horizontal lines along a row. In other words, the start and end points of a train track are $(r, c1)$ and $(r, c2)$, where r represents the row number, $c1$ represents the starting column, and $c2$ represents the ending column of the train track.

The mayor of Gridland is surveying the city to determine the number of locations where lampposts can be placed. A lamppost can be placed in any cell that is *not occupied* by a train track.

Given a map of Gridland and its k train tracks, find and print the number of cells where the mayor can place lampposts.

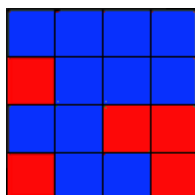
Note: A train track may overlap other train tracks within the same row.

Example

If Gridland's data is the following (1-based indexing):

```
k = 3
r  c1 c2
1  1  4
2  2  4
3  1  2
4  2  3
```

It yields the following map:



In this case, there are five open cells (red) where lampposts can be placed.

Function Description

Complete the `gridlandMetro` function in the editor below.

`gridlandMetro` has the following parameter(s):

- `int n`:: the number of rows in Gridland
- `int m`:: the number of columns in Gridland
- `int k`:: the number of tracks
- `track[k][3]`: each element contains **3** integers that represent **row**, **column start**, **column end**, all 1-indexed

Returns

- *int*: the number of cells where lampposts can be installed

Input Format

The first line contains three space-separated integers *n*, *m* and *k*, the number of rows, columns and tracks to be mapped.

Each of the next *k* lines contains three space-separated integers, *r*, *c1* and *c2*, the row number and the track column start and end.

Constraints

- $1 \leq n, m \leq 10^9$
- $0 \leq k \leq 1000$
- $1 \leq r \leq n$
- $1 \leq c1 \leq c2 \leq m$

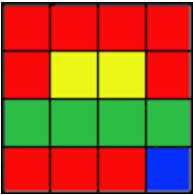
Sample Input

```
STDIN      Function
-----
4 4 3      n = 4, m = 4, k = 3
2 2 3      track = [[2, 2, 3], [3, 1, 4], [4, 4, 4]]
3 1 4
4 4 4
```

Sample Output

```
9
```

Explanation



In the diagram above, the yellow cells denote the first train track, green denotes the second, and blue denotes the third. Lampposts can be placed in any of the nine red cells.