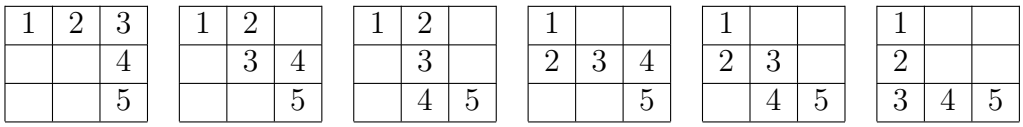
Problem B: Downright Griddy

Congratulations, your new game *Downright Griddy* has just been approved on the App Store!

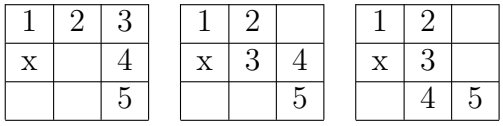
In *Downright Griddy*, you start in the top-left corner of 2D grid with r rows and c columns, with the objective of getting to the bottom-right corner of the grid. Each turn you can either move one space to the right, or one space downward. But wait, there's more! Some grid cells are blocked, and can't be walked on.

Your quality analysis team thinks that your game might be a little bit too easy, and so they want you to write a program that determines how many possible paths there are to the goal in each level. They will then somehow use this number to make your game sell more copies.

For example, here is a 3×3 grid with no blocked tiles in which there are 6 possible paths from the top-left to bottom-right:



Here is a 3×3 grid with one blocked tile in location $(2, 1)$ in which there are 3 possible paths:



Input

The first line contains the number n of test case grids in the sample input file. Each of the following n lines consists of at least 3 integers describing a test case grid to be solved. Each of these lines has the format $r\ c\ b\ r_1\ c_1\ r_2\ c_2\ \ldots r_b\ c_b$ where r and c are the numbers of rows and columns in the grid, b is the number of blocked tiles in the grid, and r_i and c_i are the row and column coordinates of the ith blocked tile (if $b > 0$). You may assume that $0 < r \leq 20$, $0 < c \leq 20$, $0 \leq b \leq r * c$, $1 \leq r_i \leq r$, $1 \leq c_i \leq c$, some input grids may have no solutions, and grid cell $(1, 1)$ in the top-left corner will never be blocked.

Output

Your program will output the number of possible paths for each test case grid, one per line.

Sample input (available as file "B.in"):

```
8
2 2 0
3 3 0
3 3 1 2 2
4 4 0
8 8 0
5 10 2 2 2 4 4
20 10 4 1 4 20 9 19 10 5 5
20 20 0
```

Sample output (available as file "B.out"):

```
2
6
2
20
3432
329
0
35345263800
```