You are given a board consists of R rows and C columns. There are N blobs in the board. Each cell of the board might or might not contain a blob. A cell can have at most 8 adjacent cells. A blob can jump over any adjacent blobs to the next empty cell. Thus a blob can jump straight to two squares at any of the 8 directions if the destination cell is empty and the cell between the source cell and the destination cell contains a blob. Blobs cannot jump outside of the board. After the jump the blob in the middle cell of the jump will be removed. The goal is to make jumps carefully so that after N-1 jumps only one blob remains. Your task is to count how many ways the goal can be achieved.

Input

Input consists of several test cases. The first line of the input file contains a single integer T indicating the number of test cases. Then T test cases follow. Each test case starts with three positive integers R, C and N. Each of the next N lines contain two integers x, y where i-th line contains the row position x and column position y of the i-th blob. No two blobs share the same cell.

Output

For each set of input produce one line of output '`Case` x`:` n', where x indicates the test case number (starting from 1) and n the number of ways the goal can be achieved.

Constraints

- $R, C \leq 4$
- $1 \leq x \leq R$
- $1 \leq y \leq C$
- $1 \leq i \leq N$

Sample Input

```
3
1 2 1
1 1
3 3 8
2 3
2 1
1 2
3 3
3 2
3 1
1 1
1 3
3 3 3
3 1
2 2
1 2
```

Sample Output

```
Case 1: 1
Case 2: 0
Case 3: 2
```