Given a grid with R rows and C columns, you are currently at $(0, 0)$ and you want to go to the position $(R-1, C-1)$. You have only two kind of movement allowed. From any position (i, j) you can go to either $(i+1, j)$ or $(i, j+1)$. You need to find the number of ways you can go to $(R-1, C-1)$ from $(0, 0)$. Easy, right? But here's is a slight problem. All the cells are not available all the time. So while counting the number of ways you need to consider that you can never step into a cell which is not available right now.

Input

First line will contain an integer T ($1 \leq T \leq 10$), which is the number of test cases. Each case starts with a line R, C and Q. Here, $1 \leq R, C \leq 1000$ and $1 \leq Q \leq 10000$. Then, Q queries follow, each with four integers a, b, c, d. This means the cells inside the rectangle with lower left corner at (a, b) and upper right corner at (c, d) are not available. All the coordinates are given in row major order with 0-based indexing. The lowermost and leftmost point is considered to be $(0, 0)$.

Output

For each case print a line '`Case` T', where T is the case number. For each query in a case, print 3 spaces and then '`Query` X`:` W', where X is query number and W is the number of ways possible for that particular query. Answer needs to be in *modulo* 912. Check sample input and output for details.

Sample Input

```
1
5 5 2
1 1 2 2
0 1 2 3
```

Sample Output

```
Case 1
   Query 1: 10
   Query 2: 5
```