

5937 Snow in Rectangleville

As you probably heard, it snows a lot in Rectangleville. The city got the strange name because of its rectangular layout and a peculiar natural phenomenon — the snowfall always occurs in a rectangular region. The meteorology department would like to keep track of how much snow fell in each part of the city. They discretized the problem by putting an $m \times n$ grid on the map of the city. They also purchased an expensive weather radar that reports the rectangle where a snowfall occurred. The output of the machine is discretized: (a) the radar reports a new rectangle for each 1 unit of snowfall, and (b) the rectangle is aligned with the grid (it is formed by a rectangular region of grid squares). All they now need is a program that will combine the outputs of the radar and compute the cumulative snowfall in each square of the grid.

Input

The input file contains several test cases, each of them as described below.

The first line contains positive integers m and n , the dimensions of the grid (m rows, n columns). The second line contains a positive integer k , the number of snowfalls reported. The next k lines contain the rectangles in which 1 unit of snowfall occurred, each rectangle is given by the coordinates of its upper-left corner square and the coordinates of its lower-right corner square. The upper-left corner square of the map has coordinates 1, 1 and the lower-right corner square of the map has coordinates m , n . (For example, if it only snows in the upper-left corner square the line would say 1 1 1 1.) You may assume that $m, n \leq 1,000$ and $k \leq 100,000$.

Output

For each test case, the output should be the cumulative snowfall (in μm) for each square of the grid, arranged in m lines (one for each row, from top to bottom), each line containing n numbers (one for each column, from left to right).

Sample Input

```
10 10
5
2 1 8 5
5 9 10 9
3 6 5 6
2 2 8 2
3 7 6 8
```

Sample Output

```
0 0 0 0 0 0 0 0 0 0
1 2 1 1 1 0 0 0 0 0
1 2 1 1 1 1 1 1 0 0
1 2 1 1 1 1 1 1 0 0
1 2 1 1 1 1 1 1 1 0
1 2 1 1 1 0 1 1 1 0
1 2 1 1 1 0 0 0 1 0
1 2 1 1 1 0 0 0 1 0
```

```
0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 1 0
```