

2060 3002 Rubbery

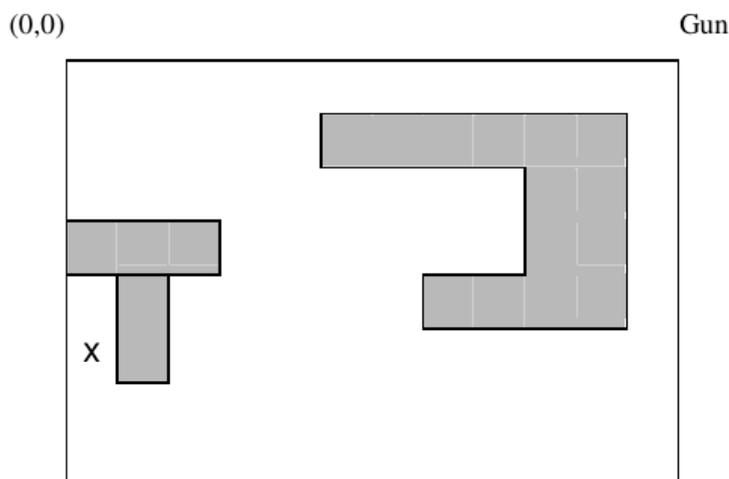
It's now 3002 and the Godfather *Unlucky Luchiano* is planning a rubbery from the Rectilinia museum. The problem is that the walls of the museum are impenetrable and the doors are guarded so his men cannot enter the museum from its doors. He is lucky that the museum has no roofs and one can enter the museum from the above. So he decides to use a device from thousands of years ago; a catapult! Using this, his men can fly and fall in some place in the museum without being caught by guards.

But another problem still exists which is a high-tech laser gun that guards the museum. The laser gun is designed based on a very recent discovery that it is possible to guide the laser beam not on a straight line, but on a rectilinear path. The fact that the beam travels the shortest (possible) distance from the emission point to a target still remains. The problem is that if one land somewhere in the museum that is reachable by the laser beams, he will be immediately destroyed. Again godfather is lucky, since in the museum, there are some walls and other obstacles through which the beam cannot pass and if his men can land in the shadows of that walls, they are safe in the way that they are not destroyed by the gun right after landing and they can use their special device to disable the laser gun. As the catapult is not a very precise device, godfather wants to know the probability of landing in shadows, so he has to compute the total area of shadowy regions of the museum.

Given the layout of the museum, you have to write a program to compute the total area of the shadowy regions. Yes! This time you really HAVE TO!

As you study the layout of the museum from the top view, you find that the museum can be considered as a rectangle with some obstacles in it. The obstacles are simple polygons with sides parallel to the rectangle sides. Interior of obstacles do not overlap. The laser gun is located in the upper-right corner of the museum. A laser beam is composed of a number of line segments; each is either horizontal or vertical. When the gun chooses its target, it intelligently determines a possible path to the target and fires. A possible path has the following characteristics:

- It consists of only horizontal and vertical segments.
- It never crosses an obstacle but in some parts may be tangent to obstacle sides. It can never be tangent to two obstacle sides at one point.
- In the travel from the gun to the target, the beam never moves from left to right or from bottom to top (directions are relative to view of the museum from the above).



The problem is to compute the total area of the shadowy regions in the museum (not belonging to the interior of the obstacles), where the laser gun cannot shoot any point inside those regions. In the above figure, the point marked by (x) is in a shadowy region (see sample input).

Input

The input file contains several test cases. The first line contains a single integer t (between 1 and 10), which is the number of test cases. Rest of the input file contains t test cases. The first line of each test case contains 2 positive integers, which are the length and the width of the rectangle respectively. The second line contains a single integer n , which is the number of obstacles in the museum (between 0 and 50). After it, there are n lines, which contain the obstacle data. Each obstacle data is in a single line, which begins with an integer m , which is the number of vertices of the obstacle (between 4 and 50), followed by $2m$ numbers, which are the x , and y coordinates of the vertices listed in clockwise order. Every coordinate in the input is a positive integer less than 1,000,000. Upper-left corner of the rectangle is the coordinate origin.

Output

The output file must contain exactly t lines. Each line contains a single number, which is the total area of the safe regions in the test case. The output for each test case is guaranteed to fit in a 32-bit integer. Total execution time for all test cases must be less than 1 minute.

Sample Input

```
1
12 8
3
8 5 1 11 1 11 5 7 5 7 4 9 4 9 2 5 2
4 0 3 3 3 3 4 0 4
4 1 4 2 4 2 6 1 6
```

Sample Output

```
12
```