2327 Farmland

We have a map for farming land in a country. The whole farming land of the country is divided into a set of disjoint farming regions. Each farmer owns only one farming region in this country. There is a boundary fence between two neighboring farming regions. The farmland map for this country can be represented in a plane graph. The following Figure-1 shows one example.

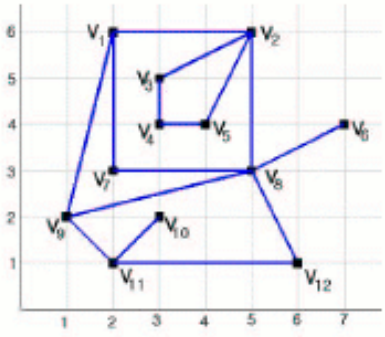

Figure-1: Farmland graph G(V,E)

There are two types of edges, boundary edge and non-boundary edge. All edges of $G(V, E)$ except (v_8, v_6) and (v_{11}, v_{10}) are boundary edges which are between two neighboring farming regions. The "proper farming region" in a Farmland graph is a closed region bounded by a simple cycle and it should not contain any vertices or edges inside. In this figure, the polygon $< v_1, v_9, v_8, v_7 >$ is a proper farming region, and the region $< v_2, v_1, v_7, v_8, v_2, v_5, v_4, v_3 >$ is not a proper farming region since its boundary cycle is not simple.

We assume that the farmland graph $G(V, E)$ is a simple connected graph, which does not allow self-loops (Figure-2 (a)) and parallel edges (Figure-2 (b)). Also in Farmland graph $G(V, E)$, we do not consider the outer face of $G(V, E)$. You can see that there are 2 proper farming regions in $G(V, E)$ shown in Figure-1, namely $< v_1, v_9, v_8, v_7 >$ and $< v_2, v_3, v_4, v_5 >$, since there are no vertices or edges inside. But the polygon $< v_1, v_7, v_8, v_2 >$ is not a proper farming region since vertex v_3, v_4, and v_5 are located in that region. Similarly, the region $< v_9, v_{11}, v_{12}, v_8 >$ is not a proper region because a vertex v_{10} is inside the region. A degenerate polygon $< v_6, v_8 >$ is not a proper region because it has no valid area inside.

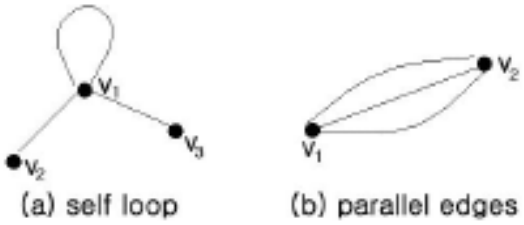

Figure-2: (a) self-loop $< v_1, v_1 >$, and (b) 3 parallel edges $\{< v_1, v_2 >, < v_1, v_2 >, < v_1, v_2 >\}$

There are other assumptions for input farmland graph data.

1. There is at least one proper farming region.
2. The position of each vertex in Farmland graph is distinct.
3. There is no edge crossing, which means the graph $G(V, E)$ is a plane graph.
4. Farmland graph $G(V, E)$ is simple and connected.

Let us define the "size" of proper farming region. The size of proper farming region is the number of boundary edges of that region. For example, the size of the proper farming region $< v_2, v_3, v_4, v_5 >$ is 4.

The problem is to find the number of proper regions that have a specified size. If you are requested to find the number of proper regions with size of 4 in the graph given in Figure-1, you must answer that there are 2 proper regions whose sizes are 4 because farming regions $< v_1, v_9, v_8, v_7 >$ and $< v_2, v_3, v_4, v_5 >$ are proper regions and their sizes are 4. If there are no such regions, then you have to print '0'.

Input

The input file consists of several test cases. The first line of the input file contains a positive integer M, the number of test cases you are to solve. After the first line, input data for M cases follow. The first line of each test case contains a positive integer N (≥ 3), the number of vertices. Each of the following N lines is of the form:

$i\ x_i\ y_i\ d_i\ a_1\ a_2\ a_3\ \ldots\ a_{d_i}$

where 'i' is the vertex number, x_i and y_i are the coordinate (x_i, y_i) of the vertex i, and d_i is the degree of the vertex i. The following $\{\ a_i\ \}$ are the adjacent vertices of the vertex i. The last line gives k, the size of proper regions that you have to count.

Note that M, the number of cases in input file is less than 10. N, the number of vertices of a given farmland graph is less than 200. All vertices are located on grid points of the 1000×1000 lattice grid.

Output

The output must contain M non-negative integers. Each line contains the answer n to the corresponding case of the input file.

Sample Input

```
2
12
1  2 6   3  9 7 2
2  5 6   4  5 3 1 8
3  3 5   2  4 2
4  3 4   2  3 5
5  4 4   2  4 2
6  7 4   1  8
7  2 3   2  8 1
8  5 3   5  7 2 9 12 6
9  1 2   3  11 8 1
10 3 2   1  11
11 2 1   3  10 9 12
12 6 1   2  8  11
4
3
1  2 2   2  2 3
2  1 1   2  1 3
3  4 1   2  1 2
4
```

Sample Output

```
2
0
```