The **ACM (Asian Cultural Museum)** authority is planning to install fire exits in its galleries in order to handle the emergency situation arising in case of a sudden fire. The museum is a collection of numerous interconnected galleries. The galleries are connected by corridors in such a way that from any gallery there is exactly one path to reach any other gallery without visiting any intermediate gallery (a gallery that is on that path) more than once.

However, in order to reduce installation cost, it has been decided that not every gallery will have a fire exit. Fire exits will be installed in such a way that if any gallery does not have a fire exit then at least one of its adjacent galleries must have one and for each corridor at least one of the two galleries it connects must have a fire exit. You are hired to determine where to put the fire exits under this constraint.

However, as a first step, you are expected to determine the minimum number of fire exits required.

Input

The input file may contain multiple test cases. The first line of each test case contains an integer N ($1 \leq N \leq 1,000$) indicating the number of galleries in this test case. Then follow N lines where the i-th ($1 \leq i \leq N$) line is the adjacency list of the i-th gallery (Each gallery is given a unique identification number from 1 to N for convenience). The adjacency list for gallery i starts with an integer n_i ($1 \leq n_i \leq N-1$) indicating the number of galleries adjacent to this gallery, followed by n_i integers giving the identification numbers of those galleries.

A test case containing a zero for N terminates the input.

Output

For each test case in the input file print a line containing the minimum number of fire exits required to meet the given constraint.

Sample Input

```
4
3 2 3 4
1 1
1 1
1 1
16
4 6 12 15 16
3 3 8 10
4 2 4 6 9
1 3
1 6
3 1 3 5
1 15
1 2
1 3
1 2
1 16
1 1
1 15
1 15
4 1 7 13 14
2 1 11
0
```

Hamiltonian path is a path in an undirected or directed graph that visits each vertex exactly once

node
1st gallery adjacent to 2 3 4
2nd --> 1
3rd --> 1
4th --> 1

Sample Output

```
1
6
```

Category: graph