# 4822   Cosmic Station

The young programmers Peter and Stancu were hired by two cosmic agencies. The agency of Peter constructed a station, composed of modules. Some couples of modules are linked by corridors in such way that it is possible to go from each module to each other module by unique path of corridors. There is neither a corridor that starts and finishes in the same module nor two or more corridors that link same two modules. *Outside* modules are linked to exactly one other module (white on the figure) and are labeled

from 1 to N. They are just for fun. All equipment of the station is concentrated in the *inner* modules — linked to more than one other module (black on the figure). The chiefs of Peter would like to keep in secret the number of inner modules. In order to hide the number of inner modules Peter encoded the topology of the station by giving for each two outside modules the distance between them (i.e. the number of corridors on the unique path between the modules).
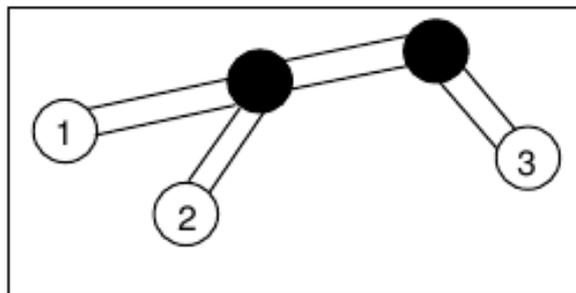
Now, Stancu has a difficult task — he promised to his bosses to decrypt the coding of Peter and to find the number of inner modules of the station. Unfortunately, he is not experienced enough. Help him. Write a program to solve the task.

Input

On the first line of the input file the number T of the test cases will be given. Each test case starts with a line containing the number N of the outside modules ($3 \le N \le 1024$). Then $N-1$ lines follow.

On the first of these lines the distances from outside module 1 to the outside modules $2, 3, \ldots, N$ are given, separated by single spaces. On the second line are given, also separated by single spaces, the distances from outside module 2 to the outside modules $3, 4, \ldots, N$, and so on. The last line contains only the distance from the outside module $N-1$ to the outside module N.

Output

For each test case the program has to print on a separate line of the standard output the number M of the inner modules of the station. In all test cases M will be less than 1024.

Sample Input

```
1
3
2 3
3
```

Sample Output

```
2
```