

Gilbert is the network admin of Ginkgo company. His boss is mad about the messy network cables on the floor. He finally walked up to Gilbert and asked the lazy network admin to illustrate how computers and switches are connected. Since he is a programmer, he is very reluctant to move throughout the office and examine cables and switches with his eyes. He instead opted to get this job done by measurement and a little bit of mathematical thinking, sitting down in front of his computer all the time. Your job is to help him by writing a program to reconstruct the network topology from measurements.

There are a known number of computers and an unknown number of switches. Each computer is connected to one of the switches via a cable and to nothing else. Specifically, a computer is never connected to another computer directly, or never connected to two or more switches. Switches are connected via cables to form a tree (a connected undirected graph with no cycles). No switches are ‘useless.’ In other words, each switch is on the path between at least one pair of computers.

All in all, computers and switches together form a tree whose leaves are computers and whose internal nodes switches (See Figure 9).

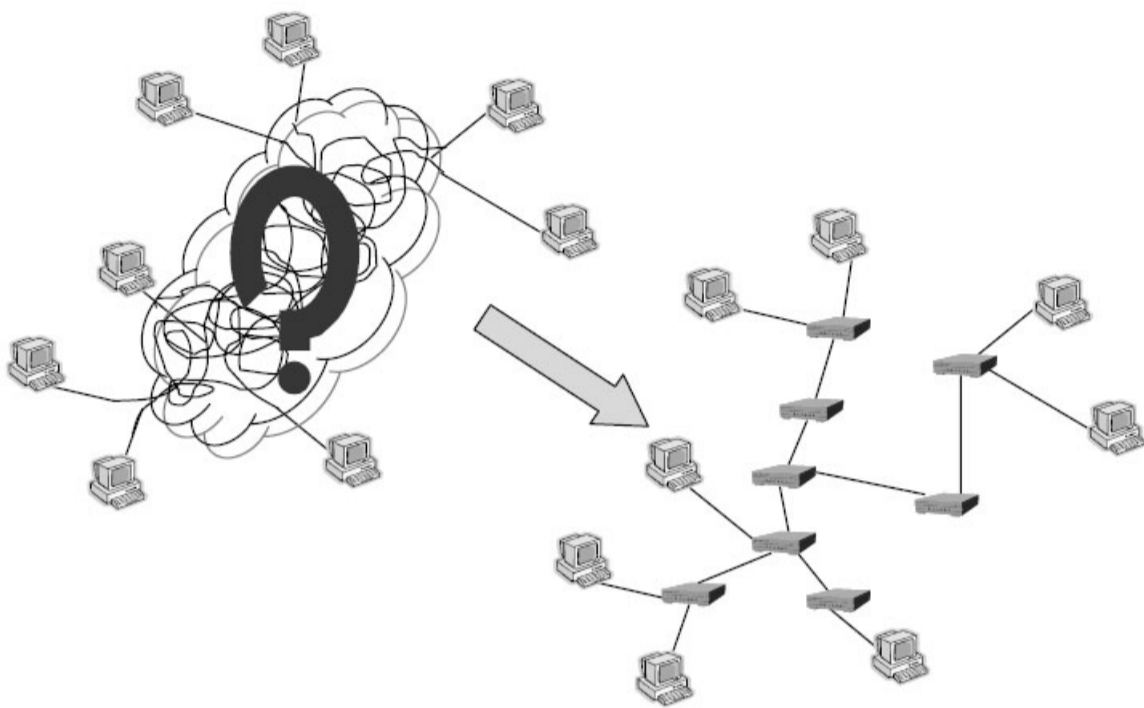


Figure 9: Computers and Switches

Gilbert measures the distances between *all pairs of computers*. The distance between two computers is simply the number of switches on the path between the two, plus one. Or equivalently, it is the number of cables used to connect them. You may wonder how Gilbert can actually obtain these distances solely based on measurement. Well, he can do so by a very sophisticated statistical processing technique he invented. Please do not ask the details.

You are therefore given a matrix describing distances between leaves of a tree. Your job is to construct the tree from it.

Input

The input is a series of distance matrices, followed by a line consisting of a single ‘0’. Each distance matrix is formatted as follows.

$$\begin{matrix}
 N \\
 a_{11} & a_{12} & \cdots & a_{1N} \\
 a_{21} & a_{22} & \cdots & a_{2N} \\
 \vdots & \vdots & \ddots & \vdots \\
 a_{N1} & a_{N2} & \cdots & a_{NN}
 \end{matrix}$$

N is the size, i.e. the number of rows and the number of columns, of the matrix. a_{ij} gives the distance between the i -th leaf node (computer) and the j -th. You may assume $2 \leq N \leq 50$ and the matrix is symmetric whose diagonal elements are all zeros. That is, $a_{ii} = 0$ and $a_{ij} = a_{ji}$ for each i and j . Each non-diagonal element $a_{ij} (i \neq j)$ satisfies $2 \leq a_{ij} \leq 30$. You may assume there is always a solution. That is, there is a tree having the given distances between leaf nodes.

Output

For each distance matrix, find a tree having the given distances between leaf nodes. Then output the degree of each internal node (i.e. the number of cables adjoining each switch), all in a single line and in ascending order. Numbers in a line should be separated by a single space. A line should not contain any other characters, including trailing spaces.

Sample Input

```

4
0 2 2 2
2 0 2 2
2 2 0 2
2 2 2 0
4
0 2 4 4
2 0 4 4
4 4 0 2
4 4 2 0
2
0 12
12 0
0

```

Sample Output

```

4
2 3 3
2 2 2 2 2 2 2 2 2 2

```