

# Catapult

*Time limit: 1 sec*

In one particular village of the people of Milagre, people do love their neighbor. There are **N** house in this village (numbered 0 to **N**-1). They love to send gift to each other very much. However, this village is out of town such that every house is very far from each other and walking to other house to deliver a gift is tiresome. So, each owner of the house sets up a catapult in the backyard and use it to fly a gift to neighbors. Catapult is very hard to adjust. When set up by a catapult specialist, each catapult can be used to send gifts to only one particular house only.

Even though they do love their neighbor, they think that it is economically unsustainable if they send gift to neighbors who won't be able to send a gift back in return. Also, it is too costly to set up additional catapult to return a gift. They would ask their neighbor to keep forwarding the gift until it reach the one they intend to send to. This is where you should help them. They want to know if they can form a group of house such that it is possible for every house in the group to send a gift, either directly or by forwarding method to other houses in the group.

Your task is to identify a largest group that each house can be it and report the number of house in each group. Note that if a house cannot send a returning gift at all, we say that this house is in a group of only one house (itself.)

## Input

- The first line of input contains one integer **N** ( $1 \leq \mathbf{N} \leq 1,000$ )
- The next **N** lines describe the catapult set up; one line per one house, starting from house 0 to house **N**-1 using the following format.
  - There are **M**+1 integers in each line. The first integer is **M** itself indicating the number of catapult in this house. The **M** following integers describe the number of house that a catapult can send a gift to.

## Output

The output must be a list of number of houses in each group. The list should be sorted ascending.

**Example**

Input	Output
3 1 1 1 2 1 0	3
3 1 1 1 0 0	1 2
5 2 1 2 2 0 3 1 3 1 2 0	1 2 2