

## Mission: Impossible

Filename: *impossible*

Time Limit: 2 seconds

You're all settled in and ready to watch the latest installment in your favorite action movie. You've got your drink, your favorite candy, but wait... You've forgotten the popcorn! Unfortunately for you, the movie theatre was as excited as you are for some spy related shenanigans, so they've installed a state of the art security system, standing between you and your beloved buttery treat. Can you deactivate the system and make it back to watch the movie?

The system consists of a series of alarms, each guarded by a series of other alarms. In order to deactivate an alarm, you must first deactivate all alarms protecting it. In addition, every alarm has an ID number associated with it. If there are multiple orderings of deactivating the alarms, you must find the lexicographically first one by ID number. For example, if both alarms 2 and 3 are unguarded, you must deactivate alarm 2 first. Since the movie theater ultimately wants to sell popcorn, it is guaranteed that there are no cycles of alarms. The theater only bought as many alarms as needed, so every alarm is at least either guarding or guarded.

### The Problem

Given an alarm configuration, output the lexicographically first sequence that deactivates all of the alarms.

### The Input

The first line of input will consist of a single positive integer,  $c$  ( $c \leq 200$ ), representing the number of input cases to process. Each case begins with a single positive integer,  $n$  ( $n \leq 50$ ), representing the number of alarms in the system. The following  $n-1$  lines will contain 2 positive integers,  $a$  and  $b$  ( $1 \leq a, b \leq n, a \neq b$ ), indicating that alarm  $a$  guards alarm  $b$ .

### The Output

For each input case, output exactly  $n$  lines, where  $n$  is the number of alarms to deactivate. Output one alarm, per line (no spaces), in the first lexicographically valid order to deactivate the alarms.

### Sample Input

```
2
5
1 2
2 3
3 4
4 5
4
2 3
1 3
4 3
```

### Sample Output

```
1
2
3
4
5
1
2
4
3
```