

# Find the Treasure

A psychic lock-lover called Digo likes playing games with Locks and Keys and also has very good logic.

One day he buys a set of **N** boxes, each of them has an index between **{1, N}** (inclusive) and no two boxes have same index. There is a key inside every box except the **Nth** box which has great treasure. He eventually finds out that due to a defect in the manufacturing of the keys, most of them could open more than one box.

The rule is that you are allowed to open only one box with any key. Each box except the first is locked. Now as Digo couldn't wait long to acquire that great treasure he requests you to find a method to open the last box starting with the key in the first box with minimum number of steps.

## INPUT

In first line, **T** no. of test cases.

For every test case:

In first line, there is an integer **N** : (number of locks)

In next **N-1** lines, on the **i'th** line there is an integer **Mi** the number of boxes which the key present in the **i'th** box can open. It is followed by the **Mi** integers (the indices of those boxes that can be opened by the key present in **i'th** box).

## OUTPUT

For every test case:

one integer **q**, minimum number of boxes opened.

In the next line : the indices of the boxes opened in order separated by space. If there are many solutions print the one which is lexicographically smallest.

If there is no way to reach the last box print "-1".

Each test case is to be followed by a blank line.

## Constraints

$1 \leq T \leq 10$

$2 \leq N \leq 100000$

$1 \leq Mi \leq 10$

## Sample Input

2

3

1 2

1 3

4

2 2 3

1 1

2 2 4

### **Sample Output**

2

1 2

2

1 3