



### Problem C Safe Networks

Input File: C.DAT

Program Source File: C.PAS or C.C or C.CPP

A server **S** of a computer network is considered critical if there are at least two other servers **A** and **B** such that **A**, **B** and **S** are different and all network interconnection paths between **A** and **B** pass through **S** (a crash of **S** disconnects **A** from **B**). An example is shown in figure 1, where the critical servers 1 and 3 are shaded. We say that a *network is safe* if it does not contain critical servers. It is known that: (1) the connection links are bi-directional; (2) a server is not directly connected to itself; (3) the network can have stand-alone sub-networks.

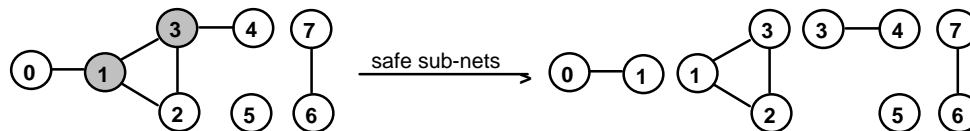


Figure 1 The maximal safe sub-nets of a network

Write a program that finds the maximal safe sub-networks of a given network. A safe sub-net is maximal if it cannot be joined with other safe sub-nets to produce a larger safe network. The program reads sets of data from a text file. Each data set specifies the structure of a network and has the format:

```
no_of_servers
server0 (no_of_direct_connections) connected_server ... connected_server
.....
serverno_of_servers-1 (no_of_direct_connections) connected_server ... connected_server
```

The first line contains an integer, **no\_of\_servers**  $\geq 0$ , which is the number of network servers. The next **no\_of\_servers** lines, one for each server in the network, are randomly ordered and show the way servers are connected. The line corresponding to **server<sub>k</sub>**,  $0 \leq k \leq \text{no\_of\_servers}-1$ , specifies the number of direct connections of **server<sub>k</sub>** and the servers that are directly connected to **server<sub>k</sub>**. Servers are represented by integers from 0 to **no\_of\_servers-1**. The items of a data line are separated by at least one space. Input data are correct. The data set from figure 2 corresponds to the network in figure 1.

input	output
8	5 sub-nets
0 (1) 1	0 1
1 (3) 2 0 3	1 2 3
2 (2) 1 3	3 4
3 (3) 1 2 4	5
4 (1) 3	6 7
7 (1) 6	
6 (1) 7	
5 (0)	

Figure 2. Input and output samples of the program

The result of the program is on standard output. For each network from the input file, the program prints the number of maximal safe sub-nets and then the servers of each sub-net, one sub-net per line. The output for the network is followed by an empty line. The servers of a sub-net are listed in ascending order and are separated by single spaces. The sub-nets are listed in ascending order according to their first component, as shown in figure 2.