

# L. Berland SU Computer Network

time limit per test: 2 seconds  
memory limit per test: 256 megabytes  
input: standard input  
output: standard output

In the computer network of the Berland State University there are  $n$  routers numbered from 1 to  $n$ . Some pairs of routers are connected by patch cords. Information can be transmitted over patch cords in both direction. The network is arranged in such a way that communication between any two routers (directly or through other routers) is possible. There are no cycles in the network, so there is only one path between each pair of routers over patch cords.

Unfortunately, the exact topology of the network was lost by administrators. In order to restore it, the following auxiliary information was collected.

For each patch cord  $p$ , directly connected to the router  $i$ , list of routers located behind the patch cord  $p$  relatively  $i$  is known. In other words, all routers path from which to the router  $i$  goes through  $p$  are known. So for each router  $i$  there are  $k_i$  lists, where  $k_i$  is the number of patch cords connected to  $i$ .

For example, let the network consists of three routers connected in chain 1 - 2 - 3. Then:

- the router 1: for the single patch cord connected to the first router there is a single list containing two routers: 2 and 3;
- the router 2: for each of the patch cords connected to the second router there is a list: one list contains the router 1 and the other — the router 3;
- the router 3: for the single patch cord connected to the third router there is a single list containing two routers: 1 and 2.

Your task is to help administrators to restore the network topology, i. e. to identify all pairs of routers directly connected by a patch cord.

## Input

The first line contains a single integer  $n$  ( $2 \leq n \leq 1000$ ) — the number of routers in the network.

The  $i$ -th of the following  $n$  lines contains a description of the lists for the router  $i$ .

The description of each list begins with the number of routers in it. Then the symbol ':' follows, and after that the numbers of routers from the list are given. This numbers are separated by comma. Lists are separated by symbol '-'.

It is guaranteed, that for each router  $i$  the total number of routers in its lists equals to  $n - 1$  and all the numbers in lists of each router are distinct. For each router  $i$  lists do not contain the number  $i$ .

## Output

Print -1 if no solution exists.

In the other case print to the first line  $n - 1$  — the total number of patch cords in the network. In each of the following  $n - 1$  lines print two integers — the routers which are directly connected by a patch cord. Information about each patch cord must be printed exactly once.

Patch cords and routers can be printed in arbitrary order.

## Examples

input
3
2:3,2
1:1-1:3
2:1,2

output
2 2 1 2 3

input
5 4:2,5,3,4 1:4-1:1-2:5,3 4:4,5,2,1 4:2,1,3,5 1:3-3:4,2,1
output
4 2 1 2 4 5 2 3 5

input
3 1:2-1:3 1:1-1:3 1:1-1:2
output
-1

Note

The first example is analyzed in the statement.

The answer to the second example is shown on the picture.

The first router has one list, which contains all other routers. The second router has three lists: the first — the single router 4, the second — the single router 1, the third — two routers 3 and 5. The third router has one list, which contains all other routers. The fourth router also has one list, which contains all other routers. The fifth router has two lists: the first — the single router 3, the second — three routers 1, 2 and 4.