

# Clockwork Bomb

Input file:            **standard input**  
Output file:          **standard output**  
Time limit:           **2.5 seconds**  
Memory limit:        **256 megabytes**

My name is James diGriz, I'm the most clever robber and treasure hunter in the whole galaxy. There are books written about my adventures and songs about my operations, though you were able to catch me up in a pretty awkward moment.

I was able to hide from cameras, outsmart all the guards and pass numerous traps, but when I finally reached the treasure box and opened it, I have accidentally started the clockwork bomb! Luckily, I have met such kind of bombs before and I know that the clockwork mechanism can be stopped by connecting contacts with wires on the control panel of the bomb in a certain manner.

I see  $n$  contacts connected by  $n - 1$  wires. Contacts are numbered with integers from 1 to  $n$ . Bomb has a security mechanism that ensures the following condition: if there exist  $k \geq 2$  contacts  $c_1, c_2, \dots, c_k$  forming a circuit, i. e. there exist  $k$  **distinct** wires between contacts  $c_1$  and  $c_2$ ,  $c_2$  and  $c_3$ ,  $\dots$ ,  $c_k$  and  $c_1$ , then the bomb immediately explodes and my story ends here. In particular, if two contacts are connected by more than one wire they form a circuit of length 2. It is also prohibited to connect a contact with itself.

On the other hand, if I disconnect more than one wire (i. e. at some moment there will be no more than  $n - 2$  wires in the scheme) then the other security check fails and the bomb also explodes. So, the only thing I can do is to unplug some wire and plug it into a new place ensuring the fact that no circuits appear.

I know how I should put the wires in order to stop the clockwork. But my time is running out! Help me get out of this alive: find the sequence of operations each of which consists of unplugging some wire and putting it into another place so that the bomb is defused.

## Input

The first line of the input contains  $n$  ( $2 \leq n \leq 500\,000$ ), the number of contacts.

Each of the following  $n - 1$  lines contains two of integers  $x_i$  and  $y_i$  ( $1 \leq x_i, y_i \leq n$ ,  $x_i \neq y_i$ ) denoting the contacts currently connected by the  $i$ -th wire.

The remaining  $n - 1$  lines contain the description of the sought scheme in the same format.

## Output

The first line should contain  $k$  ( $k \geq 0$ ) — the minimum number of moves of unplugging and plugging back some wire required to defuse the bomb.

In each of the following  $k$  lines output four integers  $a_i, b_i, c_i, d_i$  meaning that on the  $i$ -th step it is necessary to unplug the wire connecting the contacts  $a_i$  and  $b_i$  and plug it to the contacts  $c_i$  and  $d_i$ . Of course the wire connecting contacts  $a_i$  and  $b_i$  should be present in the scheme.

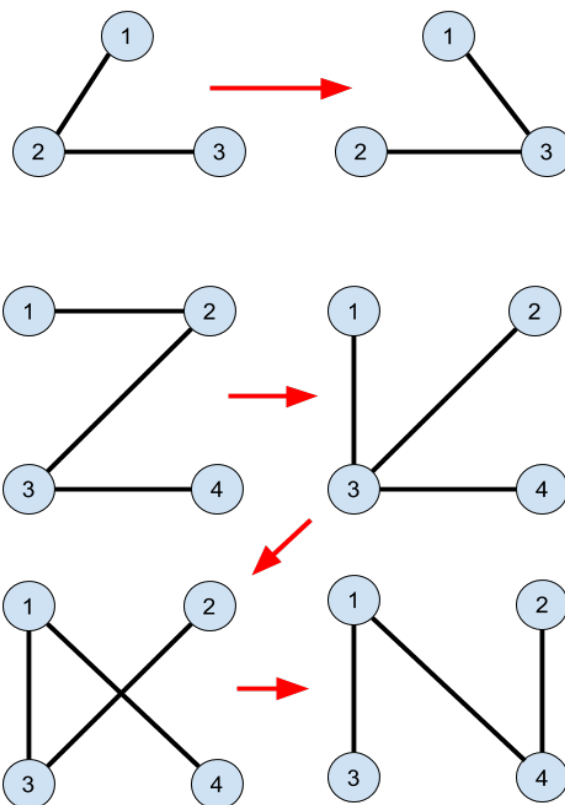
If there is no correct sequence transforming the existing scheme into the sought one, output -1.

## Examples

standard input	standard output
3 1 2 2 3 1 3 3 2	1 1 2 1 3
4 1 2 2 3 3 4 2 4 4 1 1 3	3 1 2 1 3 4 3 4 1 2 3 2 4

## Note

Picture with the clarification for the sample tests:



## Scoring

Tests for this problem are divided into six groups. For each group you earn points only if your solution passes all tests in this group and all tests in all **previous** groups.

Group	Tests	Points	Additional constraints	Comment
			$n$	
0	1 – 2	0	—	Sample tests
1	3 – 18	20	$n \leq 50$	It is guaranteed that the answer exists and requires no more than one operation
2	19 – 37	20	$n \leq 50$	
3	38 – 53	20	$n \leq 5000$	
4	54 – 69	20	$n \leq 100\,000$	
5	—	20	—	<b>Offline evaluation</b>