Problem A. Add, Remove, Transform

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

You are given a tree with n vertices. You can repeat the following operation **at most** 10^5 times.

- Choose four distinct vertices $1 \le v_1, v_2, v_3, v_4 \le n$ such that there exist edges between v_1 and v_2, v_2 and v_3, v_3 and v_4. Remove these edges and add edges between v_1 and v_3, v_1 and v_4, v_2 and v_4.

Your task is to transform the given tree so that its *diameter* is **at most** 3. Find a sequence of operations that does so.

Input

The first line of the input contains one integer n ($4 \le n \le 100$).

The i-th of the following $n-1$ lines contains two integers x_i and y_i ($1 \le x_i, y_i \le n$; $x_i \ne y_i$ for $1 \le i \le n-1$), meaning that the i-th edge connects vertices x_i and y_i in the tree.

You may assume that the given edges form a tree.

Output

At the first line, print k, the number of operations ($0 \le k \le 10^5$).

In each of the next k lines, print four integers v_1, v_2, v_3, v_4 separated by spaces. The values v_1, v_2, v_3, and v_4 should satisfy the conditions above.

If there are multiple solutions, print any one of them. It can be proven that there exists at least one way to achieve the goal.

Note that you do not have to minimize k.

Example

standard input	*standard output*
6 1 2 2 3 3 4 4 5 5 6	3 4 3 2 1 6 5 4 1 2 4 6 1

Note

The *distance* between two vertices u and v is defined as the number of edges on the unique simple path from u to v.

The *diameter* of a tree is the maximum *distance* between any two vertices.