

E. IT Restaurants

time limit per test: 1.5 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

City N. has a huge problem with roads, food and IT-infrastructure. In total the city has n junctions, some pairs of them are connected by bidirectional roads. The road network consists of $n - 1$ roads, you can get from any junction to any other one by these roads. Yes, you're right — the road network forms an undirected tree.

Recently, the Mayor came up with a way that eliminates the problems with the food and the IT-infrastructure at the same time! He decided to put at the city junctions restaurants of two well-known cafe networks for IT professionals: "iMac D0naldz" and "Burger Bing". Since the network owners are not friends, it is strictly prohibited to place two restaurants of different networks on neighboring junctions. There are other requirements. Here's the full list:

- each junction must have at most one restaurant;
- each restaurant belongs either to "iMac D0naldz", or to "Burger Bing";
- each network should build at least one restaurant;
- there is no pair of junctions that are connected by a road and contains restaurants of different networks.

The Mayor is going to take a large tax from each restaurant, so he is interested in making the total number of the restaurants as large as possible.

Help the Mayor to analyze the situation. Find all such pairs of (a, b) that a restaurants can belong to "iMac D0naldz", b restaurants can belong to "Burger Bing", and the sum of $a + b$ is as large as possible.

Input

The first input line contains integer n ($3 \leq n \leq 5000$) — the number of junctions in the city. Next $n - 1$ lines list all roads one per line. Each road is given as a pair of integers x_i, y_i ($1 \leq x_i, y_i \leq n$) — the indexes of connected junctions. Consider the junctions indexed from 1 to n .

It is guaranteed that the given road network is represented by an undirected tree with n vertexes.

Output

Print on the first line integer z — the number of sought pairs. Then print all sought pairs (a, b) in the order of increasing of the first component a .

Examples

input
5 1 2 2 3 3 4 4 5
output
3 1 3 2 2 3 1

input
10 1 2 2 3 3 4 5 6

6 7
7 4
8 9
9 10
10 4
output
6
1 8
2 7
3 6
6 3
7 2
8 1

Note
The figure below shows the answers to the first test case. The junctions with "iMac D0naldz" restaurants are marked red and "Burger Bing" restaurants are marked blue.