Problem D — limit 2 seconds

Rainbow Roads

You are given a tree with n nodes (conveniently numbered from 1 to n). Each edge in this tree has one of n colors. A path in this tree is called a *rainbow* if all adjacent edges in the path have different colors. Also, a node is called *good* if every simple path with that node as one of its endpoints is a *rainbow* path.

Find all the *good* nodes in the given tree.

A simple path is a path that does not repeat any vertex or edge.

Input

The first line of input contains a single integer n ($1 \leq n \leq 50{,}000$).

Each of the next $n-1$ lines contains three space-separated integers a_i, b_i, and c_i ($1 \leq a_i, b_i, c_i \leq n$; $a_i \neq b_i$), describing an edge of color c_i that connects nodes a_i and b_i.

It is guaranteed that the given edges form a tree.

Output

On the first line of the output, print k, the number of good nodes.

In the next k lines, print the indices of all good nodes in numerical order, one per line.

For the first sample, node 3 is good since all paths that have node 3 as an endpoint are rainbow. In particular, even though the path 3—4—5—6 has two edges of the same color (i.e. 3—4, 5—6), it is still rainbow since these edges are not adjacent.

Sample Input and Output

8 1 3 1 2 3 1 3 4 3 4 5 4 5 6 3 6 7 2 6 8 2	4 3 4 5 6

8 1 2 2 1 3 1 2 4 3 2 7 1 3 5 2 5 6 2 7 8 1	0

9 1 2 2 1 3 1 1 4 5 1 5 5 2 6 3 3 7 3 4 8 1 5 9 2	5 1 2 3 6 7

10 9 2 1 9 3 1 9 4 2 9 5 2 9 1 3 9 6 4 1 8 5 1 10 5 6 7 9	4 1 6 7 9