Rainbow Roads

Your city has decided to spice up its image – by painting its roads different colors!

Your city will paint a road the same uniform color between two intersections if there are no intersections in between (for our purposes, we'll refer to dead ends and cul-de-sacs as intersections), but along its full length, a road may be painted many different colors. Interestingly, there is exactly one path along its roads between any two intersections in the city.

The city council wants to label some intersections as *Super* intersections, and put up signs designating them so. They consider a path a *Rainbow* if there are no intersections along the path where the road in and the road out are the same color. An intersection is a *Super* intersection if the path from that intersection to every other intersection is a *Rainbow*.

Input

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs. Each test case will begin with a line of input containing a single integer n ($1 \le n \le 50{,}000$) which is the number of intersections. The intersections are numbered 1..n.

Each of the next n-1 lines will contain three integers, a, b and c ($1 \le a,b,c \le n$, $a \ne b$), which describe a road between intersection a and intersection b with color c. It is guaranteed that the given roads satisfy the constraint that there is exactly one path between any pair of intersections. The roads are two-way roads, so a road from a to b also goes from b to a.

Output

On the first line, output a single integer indicating the number of *Super* intersections. On the following lines, output a list of integers, one per line. These are the *Super* intersections. Print them in numerical order, smallest to largest.

Sample Input	Sample 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