Problem A. Tree Problem

Input file:	`stdin`
Output file:	`stdout`
Time limit:	2.5 seconds
Memory limit:	256 megabytes

You are given a tree consisting of N vertices numbered from 1 to N.

A *simple path* P between vertices a and b is a sequence of k vertices $(a = P_1, P_2, \ldots, P_k = b)$ such that any two consequent points are connected by an edge and each vertex appears in the path at most once. Note that a and b can be equal. We also say that an edge lies on a simple path if it connects two subsequent vertices of that path.

The *neighborhood* of a simple path P consists of all edges that have exactly one endpoint in P.

Each edge can be either blocked or unblocked. Initially, all edges are blocked. Write a program that handles Q queries of two types:

0. Count how many blocked edges lie on the simple path between vertices a and b.
1. Unblock all edges lying on the simple path between vertices a and b and block all edges from neighborhood of that path.

Input

The first line contains number of vertices N $(1 \le N \le 200\,000)$.

Each of the following $N - 1$ lines contains a description of an edge in the form a_i b_i. It is guaranteed that the given graph is a tree.

The next line contains number of queries Q $(1 \le Q \le 300\,000)$.

The remaining Q lines contain queries in the form t_i a_i b_i, denoting the type of the query and the endpoints respectively.

Output

For each query of type 0, output the answer to the query on a separate line.

Examples

stdin	stdout
19 1 2 2 3 1 5 5 4 5 6 6 7 6 8 1 11 11 12 11 13 11 10 10 9 13 14 13 15 15 16 15 17 15 18 15 19 6 1 19 8 0 16 2 0 16 3 1 12 9 0 19 8 0 16 9	2 3 2 2