Problem C. Steel Ball Run

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	256 mebibytes

You are given a tree with n vertices. Each vertex can contain a chip. Initially, all vertices are empty.

You have to handle two types of queries:

1. Place a chip in a vertex
2. Remove the chip from a vertex

After each query, you have to print the *span* of the current configuration of chips.

The span is defined as the minimum number of operations required to move all chips to the same vertex. In one operation, you can move a chip from its vertex to any adjacent vertex. Of course, during this process one vertex can contain multiple chips.

Note that these operations are needed only for the definition of span, and in fact, they are not performed.

Input

The first line contains a single integer n ($1 \le n \le 10^5$), the number of vertices in the tree.

The next $n-1$ lines describe tree edges, one per line. The i-th of them contains two integers u and v ($1 \le u, v \le n$), which are indices of the vertices connected by the i-th edge.

It is guaranteed that these edges form a tree.

The next line contains a single integer q ($1 \le q \le 10^5$), the number of queries.

The next q lines describe queries, one per line. Each query is described as "c v" ($1 \le v \le n$). The character c is equal to "+" for the first type of query and "−" for the second type. The integer v is the index of the vertex to which the query applies. It is guaranteed that, if the query is of the first type, there is no chip in the v-th vertex, and if the query is of the second type, there is a chip in the v-th vertex. It is also guaranteed that after each query there is at least one chip in the tree.

Output

For each query, print a line with a single integer: the span of the tree after applying this query.

Examples

standard input	standard output
3 1 2 2 3 4 + 1 + 3 + 2 - 1	0 2 2 1
6 1 2 2 3 3 4 4 5 2 6 5 + 1 + 4 + 5 - 5 + 6	0 3 4 3 4