E. Little Elephant and Tree

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The Little Elephant loves trees very much, he especially loves root trees.

He's got a tree consisting of n nodes (the nodes are numbered from 1 to n), with root at node number 1. Each node of the tree contains some list of numbers which initially is empty.

The Little Elephant wants to apply m operations. On the i-th operation $(1 \le i \le m)$ he first adds number i to lists of all nodes of a subtree with the root in node number a_i, and then he adds number i to lists of all nodes of the subtree with root in node b_i.

After applying all operations the Little Elephant wants to count for each node i number c_i — the number of integers j $(1 \le j \le n; j \ne i)$, such that the lists of the i-th and the j-th nodes contain at least one common number.

Help the Little Elephant, count numbers c_i for him.

Input

The first line contains two integers n and m $(1 \le n, m \le 10^5)$ — the number of the tree nodes and the number of operations.

Each of the following n - 1 lines contains two space-separated integers, u_i and v_i $(1 \le u_i, v_i \le n, u_i \ne v_i)$, that mean that there is an edge between nodes number u_i and v_i.

Each of the following m lines contains two space-separated integers, a_i and b_i $(1 \le a_i, b_i \le n, a_i \ne b_i)$, that stand for the indexes of the nodes in the i-th operation.

It is guaranteed that the given graph is an undirected tree.

Output

In a single line print n space-separated integers — $c_1, c_2, ..., c_n$.

Examples

input

```
5 1
1 2
1 3
3 5
3 4
2 3
```

output

```
0 3 3 3 3
```

input

```
11 3
1 2
2 3
2 4
1 5
5 6
5 7
5 8
6 9
8 10
```

```
8 11
2 9
3 6
2 8
```

output

```
0 6 7 6 0 2 0 5 4 5 5
```