F. Escape Through Leaf

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a tree with n nodes (numbered from 1 to n) rooted at node 1. Also, each node has two values associated with it. The values for i-th node are a_i and b_i.

You can jump from a node to any node in its subtree. The cost of one jump from node x to node y is the product of a_x and b_y. The total cost of a path formed by one or more jumps is sum of costs of individual jumps. For every node, calculate the minimum total cost to reach any leaf from that node. Pay attention, that root can never be leaf, even if it has degree 1.

Note that you cannot jump from a node to itself.

Input

The first line of input contains an integer n $(2 \le n \le 10^5)$ — the number of nodes in the tree.

The second line contains n space-separated integers $a_1, a_2, ..., a_n(-10^5 \le a_i \le 10^5)$.

The third line contains n space-separated integers $b_1, b_2, ..., b_n(-10^5 \le b_i \le 10^5)$.

Next $n - 1$ lines contains two space-separated integers u_i and v_i $(1 \le u_i, v_i \le n)$ describing edge between nodes u_i and v_i in the tree.

Output

Output n space-separated integers, i-th of which denotes the minimum cost of a path from node i to reach any leaf.

Examples

input

```
3
2 10 -1
7 -7 5
2 3
2 1
```

output

```
10 50 0
```

input

```
4
5 -10 5 7
-8 -80 -3 -10
2 1
2 4
1 3
```

output

```
-300 100 0 0
```

Note

In the first example, node 3 is already a leaf, so the cost is 0. For node 2, jump to node 3 with cost $a_2 \times b_3 = 50$. For node 1, jump directly to node 3 with cost $a_1 \times b_3 = 10$.

In the second example, node 3 and node 4 are leaves, so the cost is 0. For node 2, jump to node 4 with cost $a_2 \times b_4 = 100$. For node 1, jump to node 2 with cost $a_1 \times b_2 = -400$ followed by a jump from 2 to 4 with cost $a_2 \times b_4 = 100$.