

Shuttle Bus

Input file: **standard input**
Output file: **standard output**
Time limit: 2.5 seconds
Memory limit: 512 megabytes

The city contains n locations, numbered from 1 to n and connected by $n - 1$ bi-directional roads. The length of each road is 1. That is to say, the city looks like a tree in graph theory.

Nikuniku is working for the Machinery Factory and needs to choose a location t to build a new workshop in the city.

There are a_i staff living in location i .

After choosing the target location, Nikuniku will design **at most** k shuttle bus routes, to help staff commute from home to the factory. The i -th route is a simple path between location s_i and t ($s_i \neq t$) inclusive.

Each staff will walk to the nearest location where at least one route passes by and get on the bus. The capacity of the bus is large enough so don't worry about it. The walking distance of one staff is the length of the shortest path between the living location and the location getting on the bus.

Nikuniku wants to know the minimum total walking distance of all staff for every location t ($1 \leq t \leq n$), if Nikuniku chooses location t to build the workshop and designs the proper routes.

But Nikuniku just got laid off for taking too many paid leaves, so you are asked to help her by using your programming skill.

Input

The first line of the input contains two integers n ($1 \leq n \leq 2 \cdot 10^5$) and k ($1 \leq k \leq n$) — the number of locations and the number of shuttle bus routes.

The second line contains n integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^6$) — the number of staff living in each location.

Each of the next $n - 1$ lines contains two integers x and y ($1 \leq x, y \leq n, x \neq y$) - meaning that there is a road between location x and y .

It is guaranteed that these roads form a tree.

Output

Output n lines. The i -th line output an integer, minimum total walking distance if building the workshop in location i .

Examples

standard input	standard output
5 2	2
1 2 3 4 5	0
1 2	0
1 4	3
3 4	0
4 5	
8 3	3
3 1 4 1 5 9 2 6	3
2 1	1
3 1	2
5 1	3
2 7	1
2 4	1
8 5	1
6 2	