//dfs precomuting sum of a tree(basically we keep a track of parent of each node)

#include<bits/stdc++.h>

using namespace std;

vector<vector<int>>g;

long sum[100008]={0};

int data[100008];

int parent[100006];// an array to keep a track of parent of each tree

void edge(int u, int v)

{

g[u].push\_back(v);

g[v].push\_back(u);

}

int dfs(int x)

{

sum[x]=data[x];

for(int i=0;i<g[x].size();i++)

if(g[x][i]!=parent[x])

{parent[g[x][i]] =x;

sum[x]=sum[x]+dfs(g[x][i]);

}

return(sum[x]);

}

int main()

{

int n;

cin>>n;

for(int i=0;i<=n;i++)

{

parent[i]=-1;//initially we dont know anyones parents

}

g.assign(n+1, vector<int>());

for(int i=1;i<=n;i++)

{

cin>>data[i];

}

for(int i=0;i<n-1;i++)

{

int u,v;

cin>>u>>v;

edge(u,v);

}

long total= dfs(1);

long ans=LONG\_MAX;

for(int i=;i<=n;i++)

{

cout<<sum[i]<<" ";

}

return 0;

}