short

There are two types of problem statements: long ones and short ones. This one belongs to the latter.

Barr the Bear has a weighted tree: It has n vertices numbered 1 to n, $n-1$ edges, is connected, and each edge i has some positive length d_i. For each $1 \leq k \leq n$, he wants to pick some vertex v such that the total sum of distances from vertex v to vertices $1, ..., k$

$$\sum_{i=1}^{k} \text{dist}(i, v)$$

is minimized. Can you help him?

Implementation

You should submit a file that implements the following procedure:

`void shortSolver(int n, int parent[], int length[])`

You will be provided with the following procedure to use.

`void answer(long long distance)`

Your implementation of `shortSolver()` must call `answer()` exactly n times, once for each $1 \leq k \leq n$ **in order from** $k = 1$ **to** $k = n$, each time giving the minimum total distance from vertices $1, ..., k$ to some optimal vertex v.

Your procedure: shortSolver

`void shortSolver(int n, int parent[], int length[])`

Description

You will need to implement this function.

After determining the minimum total distance for each k, this procedure will need to call `answer()` to give its solution. In total, it should call `answer()` exactly n times.

Parameters

- `int n` - The number of vertices
- `int parent[]` - `parent[i]` ($2 \leq i \leq n$) gives the parent vertex of vertex i. Vertex 1 is the root.

- `int length[]` - `length[i]` ($2 \leq i \leq n$) gives the length of the edge from vertex i to the parent vertex of vertex i.

Grader procedure: answer

`void answer(long long distance)`

Description

Call this function once for each value of k from 1 to n **in order**, reporting the minimum total distance from some optimal vertex v to vertices $1, ..., k$.

Parameters

- `long long distance` - The minimum total distance from some optimal vertex v to vertices $1, ..., k$, where this is the k-th call to `answer()`.

Subtasks

For all subtasks, the length of each edge will be in the range $[1, 10^9]$.

Subtask 1 (19 points)

$1 \leq n \leq 200$.

Subtask 2 (27 points)

$1 \leq n \leq 2000$.

Subtask 3 (54 points)

$1 \leq n \leq 200000$.

Sample Input and Output

Input	Output
10 4 1 1 1 3 1 3 1 5 1 6 1 6 1 8 1 4 1	0 3 3 4 5 7 10 13 16 19
15 1 3 12 5 5 2 12 1 7 5 5 1 6 1 12 1 11 1 12 4 1 1 5 5 10 4 1 2	0 3 9 13 14 21 22 29 31 37 41 41 47 56 59