

Problem E. HDRF

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

If you love Big Data, you should be familiar with running code in a distributed manner. This always requires lots of infrastructure elements working together to make the parallel computations possible. One of such elements is usually a scheduler that decides which scheduled tasks are to be started now in some “fair” and “efficient” way.

Based on the nature of the tasks (testing, long-running, real-time, etc.), they are organized into hierarchical structure which can be represented as a rooted tree.

The following problem is inspired by one of the modern scheduling algorithms called Hierarchical Dominant Resource Fairness (HDRF).

You are given a rooted tree T with root at vertex 1 which consists of n vertices. Each vertex i of the tree gets a unique priority v_i . For each vertex, we can compute the value r_i : the smallest v_i in the subtree of vertex i including itself.

Consider the following tree traversal algorithm:

1. Start at the root vertex.
2. Choose the direct child of the current vertex which has the smallest value r_i .
3. Go to this child.
4. If the current vertex is a leaf, write it down and remove it from the tree (when we delete a vertex, we recompute all r_i). Otherwise, go to step 2.

Repeat the above procedure starting from step 1 until the tree is empty.

Given a tree T and the numbers v_i , compute the order in which vertices will be written down.

Input

The first line contains an integer n ($2 \leq n \leq 100\,000$), the number of vertices in the tree.

The second line contains $n - 1$ integers, where i -th integer p_i ($1 \leq p_i \leq n$) is the parent of vertex $(i + 1)$ in the tree. Vertices are numbered by integers from 1 to n . It is guaranteed that the input forms a valid rooted tree with root at vertex 1.

The third line contains n distinct integers v_1, v_2, \dots, v_n ($0 \leq v_i \leq 10^9$), the priorities of vertices.

Output

Output n vertices in the order they will be written down by the algorithm.

Example

standard input	standard output
5 4 4 1 1 3 5 2 1 4	3 2 4 5 1