

The cows have once again tried to form a startup company, failing to remember from past experience that cows make terrible managers!

The cows, conveniently numbered $1 \dots N$ ($1 \leq N \leq 100,000$), organize the company as a tree, with cow 1 as the president (the root of the tree). Each cow except the president has a single manager (its "parent" in the tree). Each cow i has a distinct proficiency rating, $p(i)$, which describes how good she is at her job. If cow i is an ancestor (e.g., a manager of a manager of a manager) of cow j , then we say j is a subordinate of i .

Unfortunately, the cows find that it is often the case that a manager has less proficiency than several of her subordinates, in which case the manager should consider promoting some of her subordinates. Your task is to help the cows figure out when this is happening. For each cow i in the company, please count the number of subordinates j where $p(j) > p(i)$.

Input

The first line of input contains N .

The next N lines of input contain the proficiency ratings $p(1) \dots p(N)$ for the cows. Each is a distinct integer in the range $1 \dots 1,000,000,000$.

The next $N-1$ lines describe the manager (parent) for cows $2 \dots N$. Recall that cow 1 has no manager, being the president.

Output

Please print N lines of output. The i th line of output should tell the number of subordinates of cow i with higher proficiency than cow i .

Example

Input

```
5
804289384
846930887
681692778
714636916
957747794
1
1
2
3
```

Output

```
2
0
1
0
0
```

