

E. Danil and a Part-time Job

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Danil decided to earn some money, so he had found a part-time job. The interview have went well, so now he is a light switcher.

Danil works in a rooted tree (undirected connected acyclic graph) with n vertices, vertex 1 is the root of the tree. There is a room in each vertex, light can be switched on or off in each room. Danil's duties include switching light in all rooms of the subtree of the vertex. It means that if light is switched on in some room of the subtree, he should switch it off. Otherwise, he should switch it on.

Unfortunately (or fortunately), Danil is very lazy. He knows that his boss is not going to personally check the work. Instead, he will send Danil tasks using Workforces personal messages.

There are two types of tasks:

- 1. `pow v` describes a task to switch lights in the subtree of vertex v .
- 2. `get v` describes a task to count the number of rooms in the subtree of v , in which the light is turned on. Danil should send the answer to his boss using Workforces messages.

A subtree of vertex v is a set of vertices for which the shortest path from them to the root passes through v . In particular, the vertex v is in the subtree of v .

Danil is not going to perform his duties. He asks you to write a program, which answers the boss instead of him.

Input

The first line contains a single integer n ($1 \leq n \leq 200\,000$) — the number of vertices in the tree.

The second line contains $n - 1$ space-separated integers p_2, p_3, \dots, p_n ($1 \leq p_i < i$), where p_i is the ancestor of vertex i .

The third line contains n space-separated integers t_1, t_2, \dots, t_n ($0 \leq t_i \leq 1$), where t_i is 1, if the light is turned on in vertex i and 0 otherwise.

The fourth line contains a single integer q ($1 \leq q \leq 200\,000$) — the number of tasks.

The next q lines are `get v` or `pow v` ($1 \leq v \leq n$) — the tasks described above.

Output

For each task `get v` print the number of rooms in the subtree of v , in which the light is turned on.

Example

input
4 1 1 1 1 0 0 1 9 get 1 get 2 get 3 get 4 pow 1 get 1 get 2 get 3 get 4

output

2
0
0
1
2
1
1
1
0

Note

The tree before the task pow 1.
The tree after the task pow 1.