Modification Question I
Binary Tree Operations

Problem

Given an inorder and a preorder traversal of a binary tree, construct the unique binary tree T corresponding to these traversals. The binary tree contains unique non-negative integers as keys.The program should include the following functions:

(a) `main()`: Repeatedly reads an input character from the menu list through the terminal and executes menu driven operations accordingly.

(b) `PostOrder(T)`: Prints the post order traversal of the binary tree T.

(c) `Cousin(T,a,b)`: Print 'yes' if nodes 'a' and 'b' are cousins. A pair of nodes (a, b) are cousins if they are on the same level but have different parents.

Input Format

1. The first line contains an integer $n \in [1, 10^6]$ indicating the number of nodes in the tree.
2. The second line contains a space-separated sequence of n integers representing the PREORDER traversal of the tree T with key values $\in [1, 10^6]$.
3. The third line contains a space-separated sequence of n integers representing the INORDER traversal of the tree T with key values $\in [1, 10^6]$.
4. Each subsequent line contains a character from the set {'p', 'c', 'e'}.
   - Character 'p' calls `Postorder(T)` - to print the postorder traversal of the tree.
   - Character 'c' is followed by a positive integer a and a positive integer b separated by a space. Perform `Cousin(T,a,b)` - to check if a, b are cousins.
   - Input 'e' terminates the execution of the program.

Output Format

The output of each command should be printed on a separate line.

- For option 'p', print the postorder traversal of T. Each node's value is separated by a space.
- For option 'c', print 'yes' or 'no' (small letters).

TestCase

Input1:

```
5
1 2 3 4 5
5 4 3 2 1
p
c 1 5
e
```

Output:

```
5 4 3 2 1
no
```

Input2:

```
6
4 2 1 3 6 5
1 2 4 3 5 6
p
c 1 6
e
```

Output:

```
1 2 5 6 3 4
yes
```