**#include <bits/stdc++.h>**

**using namespace std;**

**#define M 1000000007**

**struct node {**

**int data;**

**node\* left, \*right;**

**node() {**

**left = nullptr;**

**right = nullptr;**

**}**

**};**

**pair<int, int> diameter(node\* root) {**

**if (!root) {**

**return {0, 0};**

**}**

**auto p1 = diameter(root->left);**

**auto p2 = diameter(root->right);**

**int finalHeight = max(p1.second, p2.second) + 1;**

**int finalDiameter = max({p1.first, p2.first, p1.second + p2.second + 1});**

**return {finalDiameter, finalHeight};**

**}**

**void findPostOrder(int startIn, int startPre, int n, unordered\_map<int, int>&mp,**

**vector<int>&inorder, vector<int>&preorder) {**

**if(n == 0){**

**return;**

**}**

**int rootValue = preorder[startPre];**

**int rootIdx = mp[rootValue];**

**if(rootIdx != startIn) {**

**findPostOrder(startIn, startPre + 1, rootIdx - startIn, mp, inorder, preorder);**

**}**

**if(rootIdx != startIn + n - 1) {**

**findPostOrder(rootIdx + 1,**

**startPre + rootIdx - startIn + 1,**

**n - (rootIdx - startIn + 1),**

**mp,**

**inorder,**

**preorder);**

**}**

**cout<<rootValue<<" ";**

**}**

**int main() {**

**unordered\_map<int, int>mp;**

**vector<int>inorder = {6, 1, 7, 4, 10, 8, 9};**

**vector<int>preOrder = {4, 1, 6, 7, 8, 10, 9};**

**for(int i = 0; i<inorder.size();i++) {**

**mp[inorder[i]] = i;**

**}**

**findPostOrder(0, 0, inorder.size(), mp, inorder, preOrder);**

**cout<<endl;**

**return 0;**

**}**