#include <iostream>

#include <stack>

#include <unordered\_set>

using namespace std;

// Node structure for the binary tree

struct Node {

    int data;

    Node\* left;

    Node\* right;

};

// Function to create a new node

Node\* createNode(int value) {

    Node\* newNode = new Node();

    newNode->data = value;

    newNode->left = newNode->right = nullptr;

    return newNode;

}

// Function to insert a value into the binary tree

Node\* insertNode(Node\* root, int value) {

    if (root == nullptr)

        return createNode(value);

    if (value < root->data)

        root->left = insertNode(root->left, value);

    else if (value > root->data)

        root->right = insertNode(root->right, value);

    return root;

}

// Function to perform non-recursive inorder traversal

void inorderTraversal(Node\* root) {

    stack<Node\*> stk;

    Node\* current = root;

    while (current != nullptr || !stk.empty()) {

        while (current != nullptr) {

            stk.push(current);

            current = current->left;

        }

        current = stk.top();

        stk.pop();

        cout << current->data << " ";

        current = current->right;

    }

}

// Function to perform non-recursive preorder traversal

void preorderTraversal(Node\* root) {

    stack<Node\*> stk;

    stk.push(root);

    while (!stk.empty()) {

        Node\* current = stk.top();

        stk.pop();

        cout << current->data << " ";

        if (current->right != nullptr)

            stk.push(current->right);

        if (current->left != nullptr)

            stk.push(current->left);

    }

}

// Function to perform non-recursive postorder traversal

void postorderTraversal(Node\* root) {

    stack<Node\*> stk1, stk2;

    stk1.push(root);

    while (!stk1.empty()) {

        Node\* current = stk1.top();

        stk1.pop();

        stk2.push(current);

        if (current->left != nullptr)

            stk1.push(current->left);

        if (current->right != nullptr)

            stk1.push(current->right);

    }

    while (!stk2.empty()) {

        cout << stk2.top()->data << " ";

        stk2.pop();

    }

}

int main() {

    Node\* root = nullptr;

    int n, value;

    cout << "Enter the number of nodes in the binary tree: ";

    cin >> n;

    cout << "Enter the values of the nodes: ";

    for (int i = 0; i < n; ++i) {

        cin >> value;

        root = insertNode(root, value);

    }

    cout << "Non-recursive Inorder traversal: ";

    inorderTraversal(root);

    cout << endl;

    cout << "Non-recursive Preorder traversal: ";

    preorderTraversal(root);

    cout << endl;

    cout << "Non-recursive Postorder traversal: ";

    postorderTraversal(root);

    cout << endl;

    return 0;

}