Aim: To perform different traversals on a tree.

Code:

#include <stdio.h>

#include <stdlib.h>

#include <iostream>

using namespace std;

struct TreeNode {

int data;

struct TreeNode\* left;

struct TreeNode\* right;

};

struct TreeNode\* createNode(int value) {

struct TreeNode\* newNode = (struct TreeNode\*)malloc(sizeof(struct TreeNode));

if (newNode != NULL) {

newNode->data = value;

newNode->left = NULL;

newNode->right = NULL;

}

return newNode;

}

struct TreeNode\* insertNode(struct TreeNode\* root, int value) {

if (root == NULL) {

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;

}

void inOrderTraversal(struct TreeNode\* root) {

if (root != NULL) {

inOrderTraversal(root->left);

printf("%d ", root->data);

inOrderTraversal(root->right);

}

}

void postOrderTraversal(struct TreeNode\* root) {

if (root != NULL) {

postOrderTraversal(root->left);

postOrderTraversal(root->right);

printf("%d ", root->data);

}

}

void preOrderTraversal(struct TreeNode\* root) {

if (root != NULL) {

printf("%d ", root->data);

postOrderTraversal(root->left);

postOrderTraversal(root->right);

}

}

void freeTree(struct TreeNode\* root) {

if (root != NULL) {

freeTree(root->left);

freeTree(root->right);

free(root);

}

}

int main() {

struct TreeNode\* root = NULL;

int nodeValue;

char choice;

do {

printf("Input a value to insert into the binary tree: ");

scanf("%d", &nodeValue);

root = insertNode(root, nodeValue);

printf("Want to insert another node? (y/n): ");

scanf(" %c", &choice);

} while (choice == 'y' || choice == 'Y');

printf("\nIn-order Traversal of the Binary Tree: ");

inOrderTraversal(root);

printf("\n");

printf("\npost-order Traversal of the Binary Tree: ");

postOrderTraversal(root);

cout<<"\n";

printf("\npre-order Traversal of the Binary Tree: ");

preOrderTraversal(root);

freeTree(root);

return 0;

}