

**DSA Lab(Data Structured and Algorithms Lab)**

**Assignment # 11**

**Semester**: 3ndSemester

**Section**: C

**Submitted To:**

**Mr. Abdullah Shahrose**

**Submitted By:**

**Name**: Abdul Ahad

**Roll No**: 22-CS-071

**Task 1:**

#include <iostream>

using namespace std;

struct Node {

    int data;

    Node\* left;

    Node\* right;

};

Node\* createNode(int value) {

    Node\* newNode = new Node();

    if (!newNode) {

        cout << "Memory error\n";

        return NULL;

    }

    newNode->data = value;

    newNode->left = newNode->right = NULL;

    return newNode;

}

Node\* insertNode(Node\* 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(Node\* root) {

    if (root == NULL) {return;}

    inorderTraversal(root->left);

    cout << root->data << " ";

    inorderTraversal(root->right);

}

void preorderTraversal(Node\* root) {

    if (root == NULL) {return;}

    cout << root->data << " ";

    preorderTraversal(root->left);

    preorderTraversal(root->right);

}

void postorderTraversal(Node\* root) {

    if (root == NULL) { return; }

    postorderTraversal(root->left);

    postorderTraversal(root->right);

    cout << root->data << " ";

}

int main() {

    Node\* root = NULL;

    int choice, value;

    while (1) {

        cout << "1. Insertion\n";

        cout << "2. Inorder Traversal\n";

        cout << "3. Preorder Traversal\n";

        cout << "4. Postorder Traversal\n";

        cout << "5. Exit\n";

        cout << "Enter your choice: ";

        cin >> choice;

        switch (choice) {

            case 1:

                cout << "Enter value to be inserted: ";

                cin >> value;

                root = insertNode(root, value);

                break;

            case 2:

                cout << "Inorder Traversal: ";

                inorderTraversal(root);

                cout << endl;

                break;

            case 3:

                cout << "Preorder Traversal: ";

                preorderTraversal(root);

                cout << endl;

                break;

            case 4:

                cout << "Postorder Traversal: ";

                postorderTraversal(root);

                cout << endl;

                break;

            case 5:

                exit(0);

                break;

            default:

                cout << "Invalid choice\n";

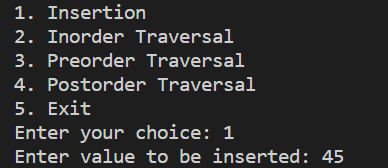
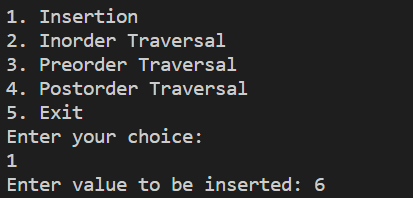
        }

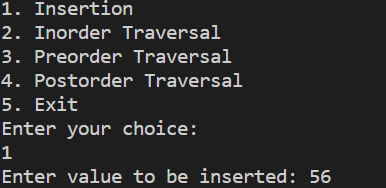
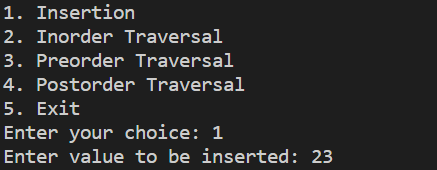
    }

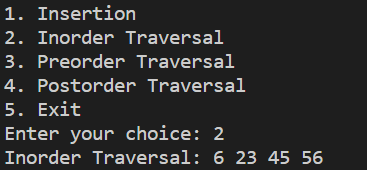
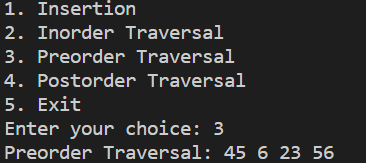
    return 0;

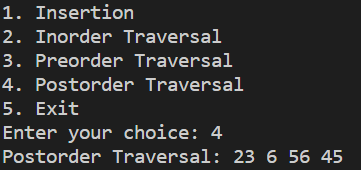
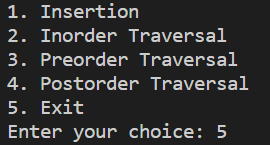
}

**Output:**

** **

** **

** **

** **