

```

#include <iostream>
#include <queue>
using namespace std;
// Binary Tree Node
struct Node {
    int data;
    Node *left, *right;
};
class tree {
public:
    Node* newNode(int data) {
        Node *node = new Node;
        node->data = data;
        node->left = NULL;
        node->right = NULL;
        return node;
    }
    Node* createRoot(int data) {
        Node *rot = newNode(data);
        return rot;
    }
    Node* leftInsert(Node* root, int data) {
        if (root == NULL) {
            return newNode(data);
        }
        root->left = leftInsert(root->left, data);
        return root;
    }
    Node* rightInsert(Node* root, int data) {
        if (root == NULL) {
            return newNode(data);
        }
        root->right = rightInsert(root->right, data);
        return root;
    }
    void preorderTraversal(Node* root) {
        if (root != NULL) {
            cout << root->data << " ";
            preorderTraversal(root->left);
            preorderTraversal(root->right);
        }
    }
    void inorderTraversal(Node* root) {
        if (root != NULL) {
            inorderTraversal(root->left);
            cout << root->data << " ";
            inorderTraversal(root->right);
        }
    }
    void postorderTraversal(Node* root) {
        if (root != NULL) {
            postorderTraversal(root->left);
            postorderTraversal(root->right);
            cout << root->data << " ";
        }
    }
};

int main() {
    tree t;
    Node *root = t.createRoot(1);
    t.leftInsert(root, 2);
    t.rightInsert(root, 3);
    t.leftInsert(root->left, 4);
    t.rightInsert(root->left, 5);
    t.leftInsert(root->right, 6);
    t.rightInsert(root->right, 7);
    cout << "Preorder traversal of the binary tree: ";
    t.preorderTraversal(root);
    cout << endl;

    cout << "Inorder traversal of the binary tree: ";
    t.inorderTraversal(root);
    cout << endl;
}

```

```
cout << "Postorder traversal of the binary tree: ";  
t.postorderTraversal(root);  
cout << endl;  
  
return 0;  
}
```