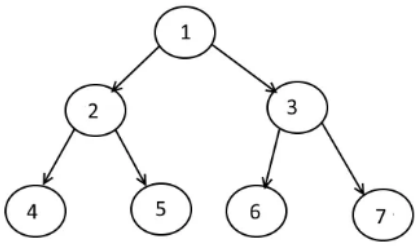
Experiment No-05: Tree Representation and Traversal C++.

Objectives

- Construct a Tree.
- Find Inorder traversal using recursion.
- Learn Level Order Traversal.

Example 1: Tree representation in C++.




```
#include<bits/stdc++.h>
using namespace std;

struct Node
{
    int data;
    Node *left; // Left reference ptr to the node.
    Node *right; // Right reference ptr to the node.

    // Method to initialize the above values.
    Node(int val)
    {
        data = val;
        left = right = NULL;
    }
};

int main()
{

   Node* root = new Node(1);
   root -> left = new Node(2);
   root -> right = new Node(3);
   root -> left -> left = new Node(4);
   root -> left -> right = new Node(5);
   root -> right -> left = new Node(6);
   root -> right -> right = new Node(7);
}
```

Example 2: Inorder traversal using recursion.

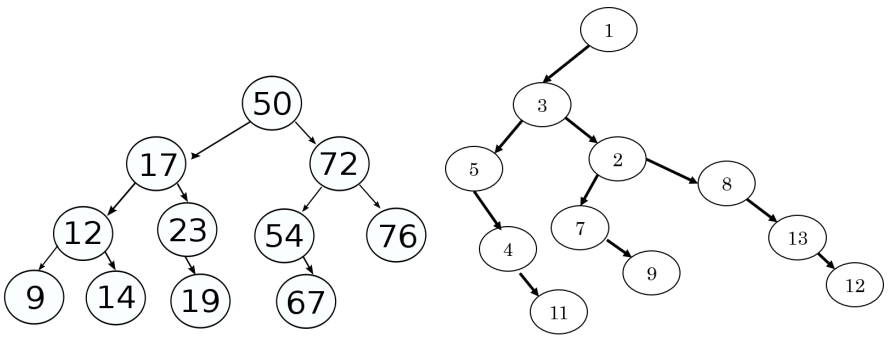
```
#include<bits/stdc++.h>
using namespace std;

struct Node
{
    int data;
    Node *left;
    Node *right;

    Node(int val)
    {
        data = val;
        left = NULL ;
        right = NULL;
    }
};

// Inorder Traversal Function

void InOrderTraversal(Node *temp)
{
    if (temp==NULL)
    {
        return;
    }

    InOrderTraversal(temp->left);
    cout<<temp->data<<" ";
    InOrderTraversal(temp->right);

}

int main()
{
   // Tree construction
   Node* root = new Node(1);
   root -> left = new Node(2);
   root -> right = new Node(3);
   root -> left -> left = new Node(4);
   root -> left -> right = new Node(5);
   root -> right -> left = new Node(6);
   root -> right -> right = new Node(7);


   court<<"Inorder Traversal:"<<endl;
   InOrderTraversal(root);

}
```

Example 2: Level-Order traversal in C++.

```
#include<bits/stdc++.h>
using namespace std;

// Level-Order Traversal Function
void LevelOrderTraversal(Node *root)
{
    if (root == NULL)
        cout<<"Tree is Empty."<<endl;

    queue<Node*> q;
    q.push(root);

    while(!q.empty()) {

        Node *temp = q.front();
        q.pop();

        if(temp->left != NULL)
            q.push(temp->left);
        if(temp->right != NULL)
            q.push(temp->right);

        cout<< temp->data<<" ";
    }

}
```

Practice Exercise

1. Write a C++ program to find the Inorder, Preorder, and Postorder traversals of the following trees.



(a) Tree 1 (b) Tree 2

Figure 1

Resources (Link)

Try to solve similar problems at an online Judge.

1. Preorder Traversal
2. Inorder Traversal
3. Postorder Traversal
4. Level Order Traversal