

## Data Structure Lab

### Assignment No-8

**Name : Sanket Shivaji Jadhav.**

**Prn: 2020BTECS00005**

#### 1. Implementation of Recursive Binary Tree Traversal.

##### Code:

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

```
class Tree
{ //Tree initiation
    public:
        Tree *right,*Left;
        int data;
        int height;
};
```

```
class Tree* createtree(){
    Tree *root=NULL;
    cout<<"Enter data\n";
    int d;
    cin>>d;

    if(d==-1) return NULL ;
    root=new Tree;//Dynamically creating a node
    root->data=d;
```

```
    cout<<"Enter the data for left node of "<<root->data<<endl;
```

```

        root->Left=createtree();
        cout<<"Enter the data for right node of "<<root-
>data<<endl;
        root->right=createtree();

        return root;
}

```

```

void preorder(class Tree* root){
    if(root==NULL) return;//if There is no further node
    than return
    // For Preorder traversal printing node is done at first
    cout<<root->data<<" ";
    preorder(root->Left);
    preorder(root->right);
}

```

```

void inorder(class Tree* root){
    if(root==NULL) return;//if There is no further node
    than return
    // For inorder traversal printing node is done in middle
    inorder(root->Left);
    cout<<root->data<<" ";
    inorder(root->right);
}

```

```

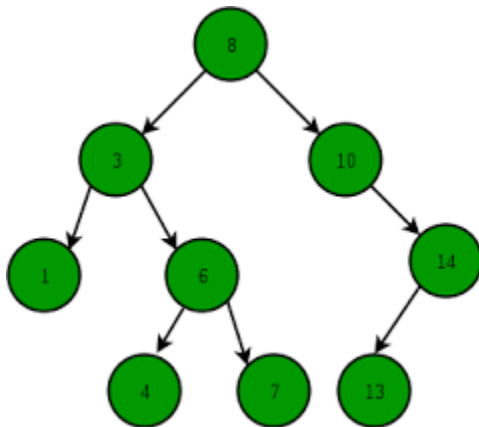
void postorder(class Tree* root){
    if(root==NULL) return;//if There is no further node
    than return
    // For inorder traversal printing node is done in middle
    postorder(root->Left);
    postorder(root->right);
}

```

```
    cout<<root->data<<" ";  
}
```

```
int main(){  
    class Tree *root=NULL;  
  
    //Creation a tree  
    root=createtree();  
  
    // Traversing a Tree  
    preorder(root);  
    inorder(root);  
    postorder(root);  
return 0;  
}
```

INPUT:



OUTPUT:



OUTPUT    DEBUG CONSOLE    TERMINAL

```
Enter the data for right node of 8
Enter data
10
Enter the data for left node of 10
Enter data
-1
Enter the data for right node of 10
Enter data
14
Enter the data for left node of 14
Enter data
13
Enter the data for left node of 13
Enter data
-1
Enter the data for right node of 13
Enter data
-1
Enter the data for right node of 14
Enter data
-1
The Preorder Traversal is: 8 3 1 6 4 7 10 14 13
The Inorder Traversal is: 1 3 4 6 7 8 10 13 14
The Postorder Traversal is: 1 4 7 6 3 13 14 10 8
```



Ad  
Go