Name-Arya Dubey

Registration Number-20BCE0908

Faculty: Professor Gopinath M.P.

Binary Tree Traversals

```
#include <stdio.h>
#include <stdlib.h>
struct node
{
    int data;
    struct node* left;
    struct node* right;
};

struct node* newNode(int data)
{
    struct node* node = (struct node*)malloc(sizeof(struct node));
    node->data = data;
    node->left = NULL;
    node->right = NULL;
     return(node);
}
 void printPostorder(struct node* node)
{
    if (node == NULL)
       return;
```

```
    printPostorder(node->left);
    printPostorder(node->right);
    printf("%d ", node->data);
}
void printInorder(struct node* node)
{
    if (node == NULL)
        return;
    printInorder(node->left);
    printf("%d ", node->data);
    printInorder(node->right);
}
void printPreorder(struct node* node)
{
    if (node == NULL)
        return;
    printf("%d ", node->data);
    printPreorder(node->left);
    printPreorder(node->right);
}
int main()
{
    struct node *root  = newNode(1);
    root->left          = newNode(2);
    root->right         = newNode(3);
    root->left->left    = newNode(4);
    root->left->right   = newNode(5);
    root->right->left   = newNode(6);
    root->right->right   = newNode(7);
    root->left->left->right    = newNode(8);
    root->left->right->right   = newNode(9);
```

```
    root->right->right->left   = newNode(10);

    printf("\nPreorder traversal of binary tree is \n");
    printPreorder(root);

    printf("\nInorder traversal of binary tree is \n");
    printInorder(root);

    printf("\nPostorder traversal of binary tree is \n");
    printPostorder(root);

    getchar();
    return 0;
}
```

```
Preorder traversal of binary tree is
1 2 4 8 5 9 3 6 7 10
Inorder traversal of binary tree is
4 8 2 5 9 1 6 3 10 7
Postorder traversal of binary tree is
8 4 9 5 2 6 10 7 3 1
```

BINARY SEARCH TREE IMPLEMENTATION

```
#include<iostream>
using namespace std;
struct node{
node *left;
int value;
node *right;
}*root=NULL;
int addnode(node *nroot,int data){
if(nroot->value==data){
return 3;
}
else{
if (nroot->value>data){
if(nroot->left!=NULL){
addnode(nroot->left,data);
}
else{
nroot->left=new node;
(nroot->left)->left=NULL;
(nroot->left)->right=NULL;
```

```
(nroot->left)->value=data;
return 1;
}
}
else{
if(nroot->right!=NULL){
addnode(nroot->right,data);
}
else{
nroot->right=new node;
(nroot->right)->left=NULL;
(nroot->right)->right=NULL;
(nroot->right)->value=data;
return 2;
}
}
}
}
void inorder(node *ptr){
if(ptr->left!=NULL)
```

```
inorder(ptr->left);
cout<<ptr->value<<" ";
if(ptr->right!=NULL)
inorder(ptr->right);
}
node* search ( int e , node* t){
 if( t != NULL )
 {
if( e == t->value )
return t;
else
 {
if( e < t -> value)
return search (e , t-> left);
else if( e > t->value )
return search (e , t -> right);
}
 }
// return NULL;
}
```

```
int main(){
int ch;
int data;
int v;
do{
cout<<"BST Implementation"<<endl;
cout<<"1. Add Node"<<endl;
cout<<"2. Inorder Travasal"<<endl;
cout<<"3. Search"<<endl;
cout<<"Enter your Choice"<<endl;
cin>>ch;
switch(ch){
case 1:
cout<<"Enter the Data to insert in the tree"<<endl;
cin>>data;
if(root==NULL){
root=new node;
root->left=NULL;
root->value=data;
root->right=NULL;
```

```
cout<<"Root node is inserted"<<endl;
}
else
v=addnode(root,data);
if(v==1){
cout<<"Node is added to left"<<endl;
}
else if(v==2){
cout<<"Node is added to right"<<endl;
}
else if(v==3){
cout<<"Dupulicate value"<<endl;
}
break;case 2:
inorder(root);
break;
case 3:
node *temp;
cout<<"Enter the Data to search in the
tree"<<endl;
```

```
cin>>data;
temp=search(data,root);
if(temp->value==data){
cout<<"Element found"<<endl;
}
else{
cout<<"Element not in the tree"<<endl;
}
break;
}
}while(1);
}
```

```
2. Inorder Travasal
3. Search
Enter your Choice
1
Enter the Data to insert in the tree
40
Node is added to right
BST Implementation
1. Add Node
2. Inorder Travasal
3. Search
Enter your Choice
1
Enter the Data to insert in the tree
50
Node is added to right
BST Implementation
1. Add Node
2. Inorder Travasal
3. Search
Enter your Choice
2
10 20 30 40 50 BST Implementation
1. Add Node
2. Inorder Travasal
3. Search
Enter your Choice
3
Enter the Data to search in the tree
55
```