Name: Abdulmuiz Khalid Shaikh
Roll no.:2101062
Subject: Data Structure and Algorithm Laboratory
Assignment No.3

```
#include<iostream>
usingnamespacestd;
//Createanodestructure
classnode
{
public:
node*left;
intdata;
node*right;
};
classBST
{
public:
node*root,*temp;
BST()
{
root=temp=NULL;
}
voidcreate();
voidinsert(node*,node*);
voidInorder(node*);
voidPreorder(node*);
voidPostorder(node*);
voidminBST();
voidmaxBST();
voidsearch(node*,intkey);
voiddescend(node*);
};
intmain()
{
BSTb;
cout<<"\nWelcometoprogram ofBST";
b.create();
cout<<"\nInorderTraversalofBSTis:-";
b.Inorder(b.root);
cout<<"\n";
cout<<"\nPreorderTraversalofBSTis:-";
b.Preorder(b.root);
cout<<"\n";
cout<<"\nPostorderTraversalofBSTis:-";
b.Postorder(b.root);
cout<<"\n";
cout<<"\nMinimum dataofBSTis:-";
b.minBST();
cout<<"\n";
cout<<"\nMaximum dataofBSTis:-";
b.maxBST();
cout<<"\n";
cout<<"\nFindingelementis:-";
b.search(b.root,2);
cout<<"\nDescendentis:-";
b.descend(b.root);
cout<<"\n";
```

```
return0;
}
voidBST::create()
{
intop,val;
do{
//1.Createanewemptynode
temp=newnode;
//2.Readadatafrom userandsaveintonode
cout<<"\nEnterdatatobesavedinnode:-";
cin>>val;
temp->data=val;
//3.MakebothptrNULL
temp->left=NULL;
temp->right=NULL;
//CreateaBST
if(root==NULL)
{
root=temp;
}
else//insertnewlycreatednodeinexistingtree
{
insert(root,temp);
}
cout<<"\nEnter1toacceptnodeagainelsepress0:-";
cin>>op;
}
while(op==1);
}
voidBST::insert(node*root,node*temp)
{
charop1;
cout<<"\nCurrentnodeis:-"<<root->data;
cout<<"\nEntertheposition(l/r):-";
cin>>op1;
if(op1=='l'||op1=='L')
{
if(root->left==NULL)
root->left=temp;
else
insert(root->left,temp);
}
elseif(op1=='r'||op1=='R')
{
if(root->right==NULL)
root->right=temp;
else
insert(root->right,temp);
}
}
voidBST::Inorder(node*temp)
{
if(temp!=NULL)
{
Inorder(temp->left);
cout<<temp->data<<"";
Inorder(temp->right);
}
}
voidBST::Preorder(node*temp)
```

```
{
if(temp!=NULL)
{
cout<<temp->data<<"";
Preorder(temp->left);
Preorder(temp->right);
}
}
voidBST::Postorder(node*temp)
{
if(temp!=NULL)
{
Postorder(temp->left);
Postorder(temp->right);
cout<<temp->data<<"";
}
}
voidBST::minBST()
{
node*temp=root;
if(temp==NULL)
{
return;
}
else
{
while(temp->left!=NULL)
{
temp=temp->left;
}
cout<<temp->data;
}
}
voidBST::maxBST()
{
node*temp=root;
if(temp==NULL)
{
return;
}
else
{
while(temp->right!=NULL)
{
temp=temp->right;
}
cout<<temp->data;
}
}
voidBST::search(node*root,intkey)
{
if(key<root->data)
{
if(root->left==NULL)
cout<<"notfound"<<endl;
elsesearch(root->left,key);
}
elseif(key>root->data)
{
if(root->right==NULL)
```

```
cout<<"notfound"<<endl;
elsesearch(root->right,key);
}
else
cout<<"found"<<endl;
}
voidBST::descend(node*temp)
{
if(temp!=NULL)
{
descend(temp->right);
cout<<temp->data<<"";
descend(temp->left);
}
}
/*
Welcometoprogram ofBST
Enterdatatobesavedinnode:-5
Enter1toacceptnodeagainelsepress0:-1
Enterdatatobesavedinnode:-62
Currentnodeis:-5
Enterthepositon(l/r):-r
Enter1toacceptnodeagainelsepress0:-1
Enterdatatobesavedinnode:-2
Currentnodeis:-5
Enterthepositon(l/r):-l
Enter1toacceptnodeagainelsepress0:-0
InorderTraversalofBSTis:-2562
PreorderTraversalofBSTis:-5262
PostorderTraversalofBSTis:-2625
Minimum dataofBSTis:-2
Maximum dataofBSTis:-62
Findingelementis:-found
Descendentis:-6252
*/
```