**INPUT**

#include<iostream>

using namespace std;

int c=0;

struct node

{ int data;

struct node \*lnode, \*rnode;

} \*root;

void inorder(struct node \*root)

{ if(root==NULL)

return;

inorder(root->lnode);

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

inorder(root->rnode); }

void preorder(struct node \*root)

{ if(root==NULL)

return;

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

preorder(root->lnode);

preorder(root->rnode); }

void postorder(struct node \*root)

{ if(root==NULL)

return;

postorder(root->lnode);

postorder(root->rnode);

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

struct node \*create()

{ int x;

struct node \*newnode=new node;

cout<<"Enter data (-1 for no node) : ";

cin>>x;

if(x==-1)

{ return 0; }

c++;

newnode->data=x;

cout<<"Enter left child of "<<x<<" | ";

newnode->lnode=create();

cout<<"Enter right child of "<<x<<" | ";

newnode->rnode=create();

return newnode; }

void display()

{ struct node \*ptr=new node;

struct node \*left1=new node;

struct node \*right1=new node;

if(root==NULL)

{    cout<<"\nBinary Search Tree is empty!\n";   }

else

{

ptr=root;

cout<<"\nBinary Search Tree:\n";

cout<<" "<<ptr->data<<"\n";

cout<<" "<<ptr->lnode->data<<"\t "<<ptr->rnode->data<<"\n";

left1=ptr->lnode;

right1=ptr->rnode;

cout<<" "<<left1->lnode->data<<" "<<left1->rnode->data<<" "<<

right1->lnode->data<<" "<<right1->rnode->data<<"\n";   } }

void smallest()

{

if(root==NULL)

{ cout<<"\nNo nodes are there\n"; }

else

{

struct node \*ptr,\*result;

ptr=root;

while(ptr!=NULL)

{

result=ptr;

ptr=ptr->lnode;

}

cout<<"\nSmallest node in BST is "<<result->data<<endl;

}

}

void largest()

{

if(root==NULL)

{ cout<<"\nNo nodes are there\n"; }

else

{

struct node \*ptr,\*result;

ptr=root;

while(ptr!=NULL)

{

result=ptr;

ptr=ptr->rnode;

}

cout<<"\nLargest node in BST is "<<result->data<<endl;

}

}

int main()

{

cout<<"BST implementation for level 2\n\n";

root=NULL;

root=create();

cout<<"\nBST created successfully\n";

int n;

do{

cout<<"\n1.Display BST\n2.Inorder traversal\n3.Preorder traversal\n4.Postorder traversal\n5.Count nodes\n6.Smallest node\n7.Largest node\n8.Exit\nEnter your choice : ";

cin>>n;

switch(n)

{

case 1:

display();

break;

case 2:

if(root==NULL)

cout<<"\nTree is empty\n";

else{ cout<<"\nIn order expression is : ";

inorder(root);

cout<<endl;}

break;

case 3:

if(root==NULL)

cout<<"\nTree is empty\n";

else{ cout<<"\nPre order expression is : ";

preorder(root);

cout<<endl;}

break;

case 4:

if(root==NULL)

cout<<"\nTree is empty\n";

else{ cout<<"\nPost order expression is : ";

postorder(root);

cout<<endl;}

break;

case 5:

cout<<"\nTotal no. of nodes is "<<c<<endl;

break;

case 6:

smallest();

break;

case 7:

largest();

break;

case 8:

break;

default:

cout<<"\nEnter from above options!!!\n";

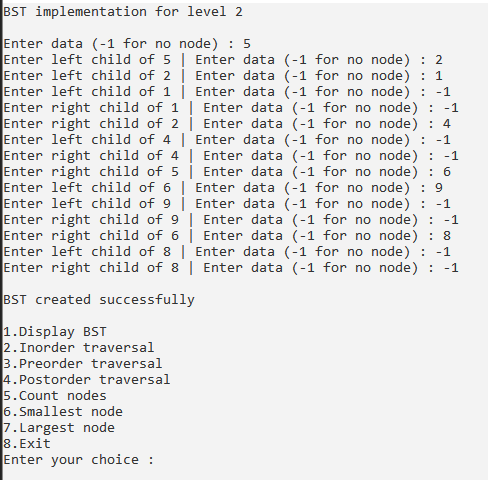
}

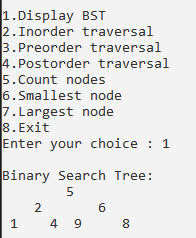
}while(n!=8);

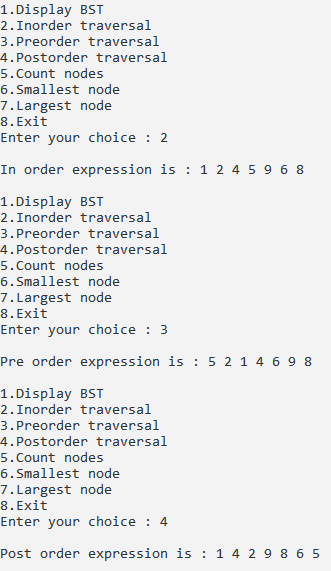
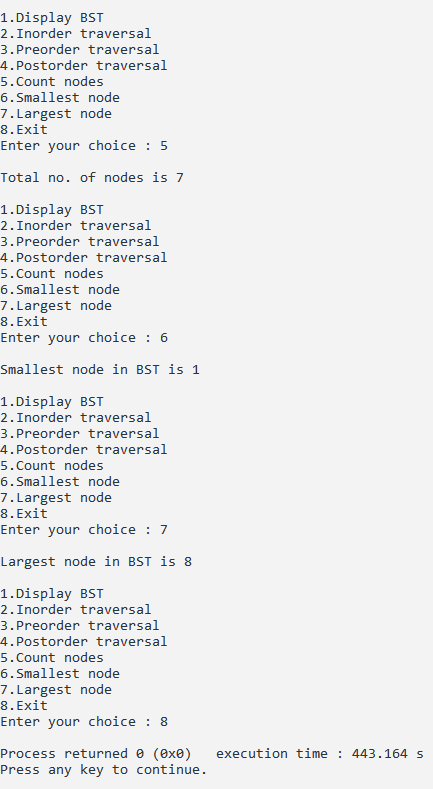
return 0;

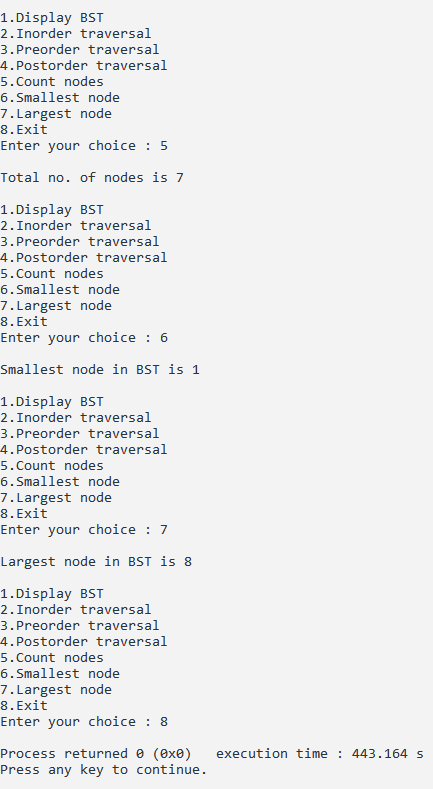
}

**OUTPUT**

****

****

** **

****