Lab Exam

DATA STRUCTURE

**Course Instructor**

Mr. SIR FAZLI BASIT

**Submitted By:**

UMAR HAYAT(P11-6143)

Date:

03 JAN, 2013

FALL 2012



**Department of Computer Sciences**

**FAST – National University of Computer & Emerging Sciences**

**Peshawar Campus**

**Task # 01:**

**Solution:**

#include <iostream>

using namespace std;

int firstloop=0;

int equalsize=0;

void bubblesearch(int [],int);

void main()

{

int \*sequence;

int n=0;

int size=0;

cout<<"ENTER your size : ";

cin>>size;

sequence=new int[size];

for(int i=0;i<size;i++)

{

cout<<"ENTER YOUR Number : ";

cin>>sequence[i];

}

equalsize=size;

bubblesearch(sequence,0);

cout<<"The sorted list is : "<<endl;

for(int i=0;i<size;i++)

cout<<"\t"<<sequence[i];

cout<<endl;

system("pause");

}

void bubblesearch(int sel[],int start)

{

if(firstloop==equalsize)

return;

if(start==equalsize)

{

firstloop++;

start=firstloop;

}

if(sel[firstloop]>sel[start])

{

int temp=sel[firstloop];

sel[firstloop]=sel[start];

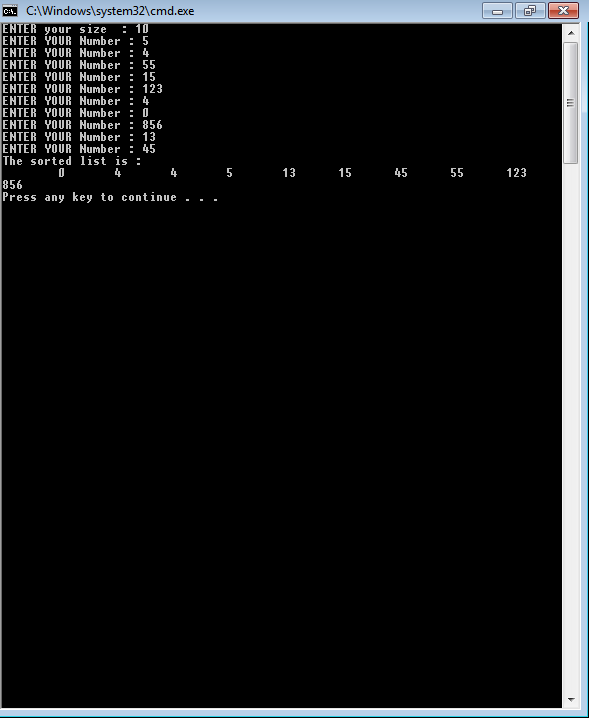
sel[start]=temp;

}

bubblesearch(sel,start+1);

}

**Screen shorts:**



**Task # 02:**

**Solution:**

#include <iostream>

#include <math.h>

#include <conio.h>

using namespace std;

int my;

int nodes=0;

struct node

{

int info;

node \*left,\*right;

node()

{

info=0;

left=right=NULL;

}

};

//

class tree

{

private:

int h;

bool flag,show;

node \*root,\*temp1,\*temp2;

public:

void insert(node\*t,int key);

node \*getroot();

void insert\_node(node \*p,int level);

void get(node \*,int key);

void height(node \*t);

void inorder(node \*p);

int heightofnode(node \*,int key);

tree()

{

flag=true;

h=0;

root=temp2=temp1=NULL;

}

};

//

void main()

{

tree t1;

int num;

menu:

cout<<"1.To insert a new node :"<<endl;

cout<<"2.Print by inorder traversal :"<<endl;

cout<<"3.TO find the height of a node :"<<endl;

cout<<"4.To exit the program : ";

cin>>num;

cout<<endl;

switch(num)

{

case 1:

cout<<"ENTER YOPUR NUMBER : ";

cin>>num;

t1.insert(t1.getroot(),num);

cout<<endl;

goto menu;

case 2:

cout<<"\tTHE INORDER TRAVERSAL IS :\n\t\t ";

t1.inorder(t1.getroot());

cout<<endl;

goto menu;

case 3:

cout<<"ENTER YOPUR NUMBER : ";

cin>>num;

cout<<"THE HEIGHT IS : "<<t1.heightofnode(t1.getroot(),num);

cout<<endl;

goto menu;

case 4:

return;

}

system("pause") ;

}

//

void tree::insert(node \*t, int key)

{

if(root==NULL)

{

root=new node();

root->info=key;

nodes++;

return;

}

h=0;

height(root);

flag=true;

my=1;

insert\_node(root,h);

temp2=temp1;

if(flag)

{

temp2->left=new node();

temp2->left->info=key;

nodes++;

return;

}

else

{

temp2->right=new node();

temp2->right->info=key;

nodes++;

flag=true;

return;

}

}

//

void tree::insert\_node(node \*p,int level)

{

if(p==NULL)

return;

if(nodes==int(pow(2.0,h+1))-1)

{

while(p->left!=NULL)

p=p->left;

temp1=p;

return;

}

if ((p->left==NULL))

{

my=2;

temp1=p;

return;

}

if ((p->right==NULL))

{

temp1=p;

my=2;

flag=false;

return;

}

else

{

if((level>1))

insert\_node(p->left,level-1);

if((level>1)&&(my==1))

insert\_node(p->right,level-1);

}

}

//

node \*tree::getroot()

{

return root;

}

//

void tree::inorder(node \*p)

{

if(p==NULL)

return;

if(p->left!=NULL)

inorder(p->left);

cout<<p->info<<" ";

if(p->right!=NULL)

inorder(p->right);

}

//

void tree::height(node \*t)

{ h=0;

if(t==NULL)

return;

if(t->left!=NULL)

{

height(t->left);

h++;

}

}

void tree::get(node \*tmp,int key)

{

if(tmp==NULL)

return;

if(tmp->info==key)

{temp2=tmp;

show=true;}

get(tmp->left,key);

get(tmp->right,key);

}

int tree:: heightofnode(node \*temp,int key)

{

int a,b,c;

show=false;

get(temp,key);

if(show==true)

{

height(temp2);

a=h;

height(temp);

b=h;

c=b-a;

return c;

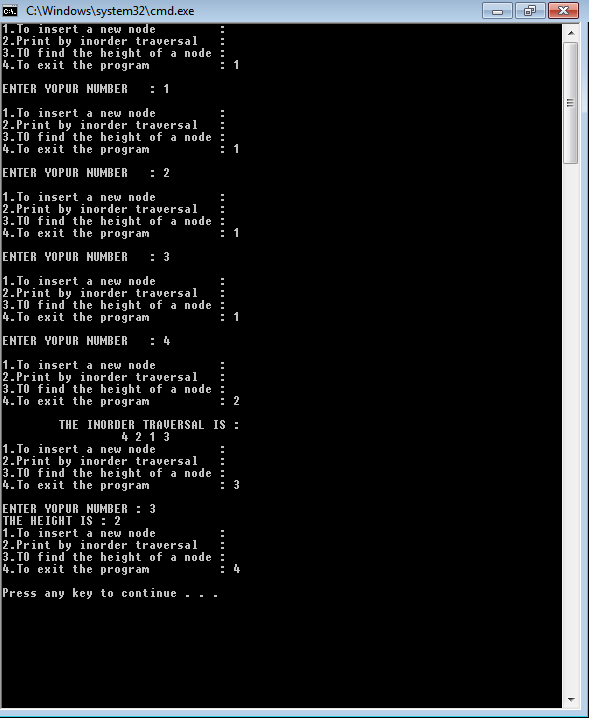
}

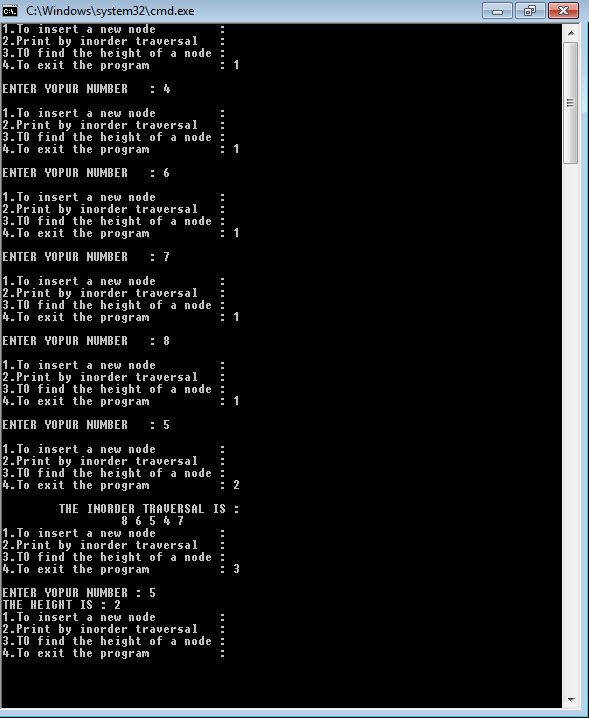
else

cout<<"THE number is not present in the tree "<<endl;

}

**Screen shorts:**

****

****

Thank u

The end