#include<iostream>

#include<string>

using namespace std;

struct arr{

int d;};

struct list

{

int dd;

list\* r;

list\* l;

};

list\* prnt=NULL;

arr arra[500];

void insertarr(int x)

{

int i=0;

while(true)

{

if(arra[i].d==-1)

{

arra[i].d=x;

break;

}

else

if(arra[i].d>x)

{

i=i\*2+1;

}

else i=i\*2+2;

}

}

void insertlist(int x)

{

list\* y=new list;

y->dd=x;

y->r=NULL;

y->l=NULL;

list\* t=prnt;

if(t==NULL)

{

prnt=y;

}

else

while(true)

{

if(t->dd>x)

{

if(t->l==NULL)

{

t->l=y;

break;

}

else t=t->l;

}

else

{

if(t->r==NULL)

{

t->r=y;

break;

}

else t=t->r;

}

}

}

void inorderarr(int z)

{

if(arra[z].d==-1)

return;

else

{

inorderarr(2\*z+1);

cout<<arra[z].d<<endl;

inorderarr(2\*z+2);

}

}

void searchlist(int x)

{

list\* t=prnt;

int f=0;

if(t==NULL)

cout<<"List is empty\n";

else

{

while(t!=NULL)

{

if(t->dd==x)

{

cout<<"found\n";

f=1;

break;

}

else if(t->dd>x) t=t->l;

else t=t->r;

}

if(f==0) cout<<"Not found\n";

}

}

void searcharr(int x)

{

int i=0;

int f=0;

if(arra[i].d==-1)

cout<<"tree is empty\n";

else

{

while(arra[i].d!=-1)

{

if(arra[i].d==x)

{

cout<<"found at\t"<<i<<endl;

f++;

break;

}

else if(arra[i].d>x) i=i\*2+1;

else i=i\*2+2;

}

if(f==0)

cout<<"not found"<<endl;

}

}

void inorderlist(list\* root)

{

if(root==NULL)

return;

else

{

inorderlist(root->l);

cout<<root->dd<<endl;

inorderlist(root->r);

}

}

void main()

{

for(int i=0; i<500; i++)

arra[i].d=-1;

int op;

int x,y;

cout<<"Enter 1 to do BST with array\nEnter 2 to do BST with linked list\nEnter 3 to exit\n";

cin>>op;

switch(op)

{

case 1:

do

{

cout<<"Enter 1 to insert node\nEnter 2 to display in inorder\nEnter 3 to search\nEnter 4 to exit\n";

cin>>x;

if(x==4)break;

switch(x)

{

case 1:

cout<<"Enter number to input\n";

cin>>y;

insertarr(y);

break;

case 2:

cout<<"Inorder\n";

inorderarr(0);

break;

case 3:

cout<<"Enter number to search\n";

cin>>y;

searcharr(y);

break;

default: cout<<"Invalid option\n";

}

}while(true);

break;

case 2:

do

{

cout<<"Enter 1 to insert node\nEnter 2 to display in inorder\n Enter 3 to Search\n 4 to Exit\n";

cin>>x;

if(x==3)break;

switch(x)

{

case 1:

cout<<"Enter number to input\n";

cin>>y;

insertlist(y);

break;

case 2:

cout<<"Inorder\n";

inorderlist(prnt);

break;

case 4:

cout<<"Enter number to searc\n";

cin>>y;

searchlist(y);

break;

default: cout<<"Invalid option\n";

}

}while(true);

}

}