#include <iostream>

class Simpul

{

public:

int iData;

double dData;

Simpul\*pAnakKiri;

Simpul\*pAnakKanan;

Simpul(): iData(0), dData(0.0), pAnakKiri(NULL), pAnakKanan(NULL){}

~Simpul()

{

cout<<"X-"<<iData<<"";

}

void tampilSimpul()

{

cout<<"{"<<IData<<"," << dData<<"}";

}

};

class Pohon

{

private:

Simpul\*pAkar;

public:

Pohon(): pAkar(NULL){

}

Simpul\*cari(int kunci)

{

Simpul\*pSkrg = pAkar;

while(pSkrg->iData != kunci)

{

if(kunci< pSkrg->iData)

pSkrg = pSkrg->pAnakKiri;

else

pSkrg = pSkrg->pAnakKanan;

if(pSkrg == NULL)

return NULL;

}

return pSkrg;

}

void sisip(int id, double dd)

{

Simpul\*SimpulBaru = New Simpul;

SimpulBaru->iData = Id;

SimpulBaru->dData = dd;

if(pAkar == NULL)

pAkar = SimpulBaru;

else

{

Simpul\*pSkrg = pAkar;

Simpul\*pInduk;

while(true)

{

pInduk = pSkrg;

if(id < pSkrg-> iData)

{

pSkrg = pSkrg->pAnakKiri;

if(pSkrg == NULL)

{

pInduk->pAnakKiri = SimpulBaru;

return;

}

}

else

{

pSkrg = pSkrg->pAnakKanan;

if(pSkrg == NULL)

{

pInduk->pAnakKanan = SimpulBaru;

return;

}

}

}

}

}

void jelajah(int tipeJelajah)

{

switch(tipeJelajah)

{

case 1: cout<<"\nPenjelajahan Preorder:";

preOrder(pAkar);

break;

case 2: cout<<"\nPenjelajahan Inorder:";

inOrder(pAkar);

break;

case 3: cout<<"\Penjelajahan Postorder:";

postOrder(pAkar);

break;

}

cout<<endl;

}

void preOrder(Simpul\*pAkarLokal)

{

if(pAkarLokal != NULL)

{

cout<<pAkarLokal->iData<<"";

preOrder(pAkarLokal->pAnakKiri);

preOrder(pAkarLokal->pAnakKanan);

}

}

void inOrder(Simpul\*pAkarLokal)

{

if(pAkarLokal != NULL)

{

inOrder(pAkarLokal->pAkarkiri);

cout<< pAkarLokal-> iData<<"";

inOrder(pAkarLokal->pAkarKanan);

}

}

void PostOrder(SImpul\*pAkarLokal)

{

if(pAkarLokal != NULL)

{

postOrder(pAkarLokal->pAnakKiri);

postOrder(pAkarLokal->pAnakKanan);

cout<<pAkarLokal->iData<<"";

}

}

void tampilPohon()

{

stack<Simpul\*>tumpukan;

tumpukan.push(pAkar);

int jumSpasi = 32;

bool apaBarisKosong = false;

cout<<"-----------------------";

cout<<endl;

while(apaBarisKosong==false)

{

stack<Simpul\*> tumpukanLokal;

apaBarisKosong = true;

for(int j=0; j<jumSpasi; j++)

cout<<"";

while(tumpukan.empty()==false)

{

Simpul\*temp=tumpukan.top();

tumpukan.pop();

if(temp != NULL)

{

cout<<temp->iData;

tumpukanLokal.push(temp->pAnakKiri);

tumpukanLokal.push(temp->pAnakKanan);

if(temp->pAnakKiri != NULL || temp->pAnakKanan != NULL)

apaBarisKosong = false;

}

else

{

cout<<"\_";

tumpukanLokal.push(NULL);

tumpukanLokal.push(NULL);

}

for(int j=0; j<jumSpasi\*2-2; j++)

cout<<"";

}

cout<<endl;

jumSpasi /= 2;

while(tumpukanLokal.empty()==false)

{

tumpukan.push(tumpukanLokal.top());

tumpukanLokal.pop();

}

}

cout<<"----------------------";

cout<<endl;

}

void hapus()

{

hapusRekursif(pAkar);

}

void hapusRekursif(Simpul\*pAkarLokal)

{

if(pAkarlokal 1+ NULL)

{

hapusRekursif(pAkarLokal->pAnakKiri);

hapusRekursif(pAkarLokal->pAnakKanan);

delete pAkarLokal;

}

}

}

int main() {

int nilai;

cahr pilihan = NULL;

Simpul\*ditemukan;

Pohon pohon;

pohon.sisip(50,5.0);

pohon.sisip(25,2.5);

pohon.sisip(75,7.5);

pohon.sisip(12,1.2);

pohon.sisip(37,3.7);

pohon.sisip(43,4.3);

pohon.sisip(30,3.0);

pohon.sisip(33,3.3);

pohon.sisip(87,8.7);

pohon.sisip(93,9.3);

pohon.sisip(97,9.7);

while(pilihan 1="k")

{

cout<<"masukan huruf pertama dari ";

cout<<"tampil, sisip, cari, jelajah, atau keluar: ";

cin>>pilihan;

switch(pilihan)

{

case 't':

pohon.tampilPohon();

break;

case 's':

cout<<"masukan nilai yang akan disisipkan: ";

cin>> nilai;

pohon.sisip(nilai, nilai+0.9);

break;

case 'c':

cout<<"masukan nilai yang akan dicari: ";

cin>> nilai;

ditemukan = pohon.cari(nilai);

if(ditemukan !=NULL)

{

cout<<"ditemukan: ";

ditemukan->tampilSimpul();

cout<<endl;

}

else

cout<<"tidak ditemukan" <<nilai<<endl;

break;

case 'k':

pohon.hapus();

cout<<endl;

break;

default;

cout<<"entri tak-valid\n";

}

}

}