using namespace std;

#include <iostream>

struct node {

int val;

node \* l, \*r;

};

typedef node\* BTree;

BTree b = 0;

void insertNode(BTree &b, int n) //حشر عقدة جديدة إلى الشجرة

{

if (b == 0)

{

b = new node();

b->val = n; b->l = 0; b->r = 0;

}

else

if (n<b->val)

insertNode(b->l, n);

else

insertNode(b->r, n);

}

/\*..................................................................\*/

void printInfix(BTree b) // طباعة عقد الشجرة حسب التجول النظامي

{

if (b != 0)

{

printInfix(b->l);

cout << b->val << " ";

printInfix(b->r);

}

}

void printPrefix(BTree b) //طباعة عقد الشجرة حسب التجول المصدر

{

if (b != 0)

{

cout << b->val << " ";

printPrefix(b->l);

printPrefix(b->r);

}

}

void printPostfix(BTree b) //طباعة عقد الشجرة حسب التجول المحلق

{

if (b != 0)

{

printPostfix(b->l);

printPostfix(b->r);

cout << b->val << " ";

}

}

/\* .................................................................\*/

void printInternal(BTree b) //طباعة العقد الداخلية

{

if (b != 0 && (b->l != 0 || b->r != 0)) {

cout << b->val << " ";

printInternal(b->l);

printInternal(b->r);

}

}

/\*..................................................................\*/

void printLeaf(BTree b) //طباعة الاوراق

{

if (b != 0) {

if (b->l == 0 && b->r == 0)

cout << b->val << " ";

else {

printLeaf(b->l);

printLeaf(b->r);

}

}

}

/\*...................................................................\*/

void Printpath(BTree b, int n) //طباعة مسار عقدة

{

BTree t = b;

while (t != 0 && t->val != n) {

cout << t->val << " ";

if (n<t->val) t = t->l;

else t = t->r;

}

if (t != 0) cout << t->val << endl;

else cout << "the node " << n << "doesn't exist .";

}

/\*...................................................................\*/

int nodeNum(BTree b) //حساب عدد العقد الكلي في الشجرة

{

if (b == 0)

return 0;

else

return 1 + nodeNum(b->l) + nodeNum(b->r);

}

/\*...................................................................\*/

void main() {

char n;

do {

cout << "Enter your choice..\n";

std::cin >> n;

switch (n) {

case '1': {

cout << "Enter node value to insert..\n";

int e;

cin >> e;

insertNode(b, e);

cout << "node " << e << " is inserted..\n";

break;

}

case '2': { printInfix(b);break;}

case '3': { printPrefix(b); break;}

case '4': { printPostfix(b); break; }

case '5': { printInternal(b); break; }

case '6': { printLeaf(b); break; }

case '7': {

cout << "Enter node value to get it path..\n";

int e;

cin >> e;

Printpath(b, e);

break;

}

case '8': { cout<<nodeNum(b); break; }

}

cout << "\n";

} while (n != 'q');

}

/\*....................................

1. حساب ارتفاع الشجرة

2. حساب ارتفاع العقدة

3.التحقق من الشجرة هل خطية ام لا

4ز التحقق من ان الشجرة تامة

5.طباعة عدد الأوراق

6.حشر عقدةفي الشجرة

7.حذف عقدة من شجرة

8.البحث عن عقدة

9.طباعة عدد العقد الداخلية

10.التأكد من أن شجرة الأولى جزء من الشجرة الثانية

11.التأكد من أن شجرة أولى هي معكوس شجرة ثانية

12.حساب حجم الشجرة

\*/