Лабораторная работа №3

“Поиск по совпадению в бинарном дереве”

Вариант 9

**1. Задача**

Написать алгоритм реализующий поиск в бинарном дереве(всех записей с ключом, равным аргументу)

**2. Общее описание работы программы**

Пользователь вводит число узлов дерева и потом вводит их по одному. В это время функция void push(int a, node \*\*t) формирует из введённых данных дерево на основе структуры

struct node {

int info; // Информационное поле

node \*l

node \*r; // Левая и Правая часть дерева

};

С помощью функции void print(node \*t, int u) выодим получившееся дерево.

Потом просим пользователя ввести узел который требуется найти в данном дереве. Заданный узел передаётся в функцию int find1(node \*t,int d) где обходом по дереву ищется заданный узел. Результат функции возвращается переменной res. Если res = 0, то такого узла нет в дереве, если res = n; то данный узел встречается в дереве n раз.

**3. Описание структур данных**

struct node {

int info;

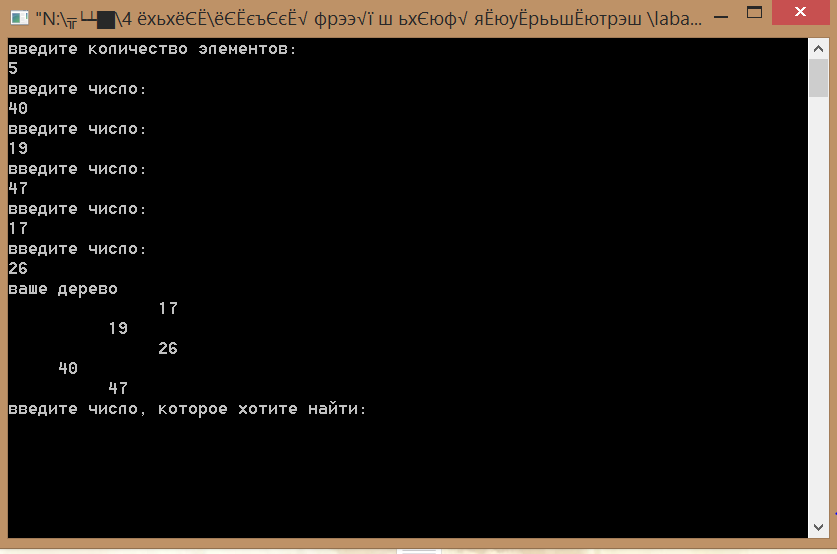
node \*l

node \*r;

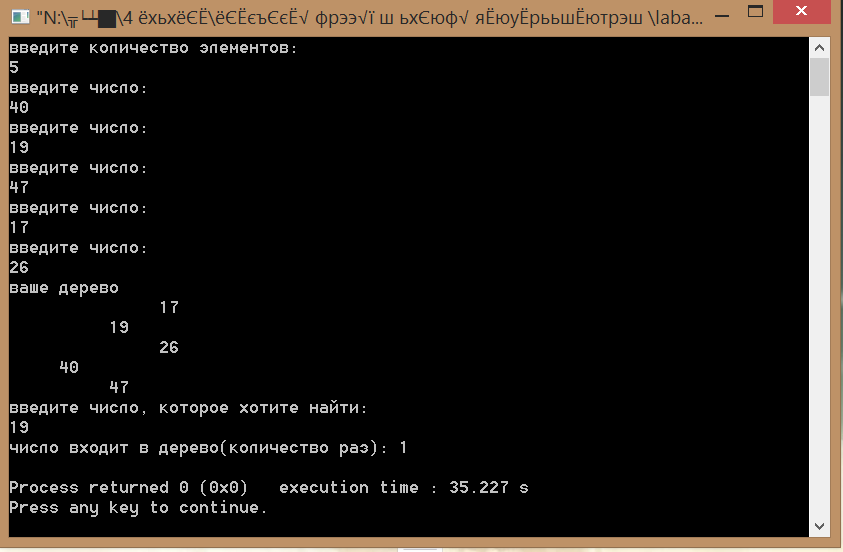
};

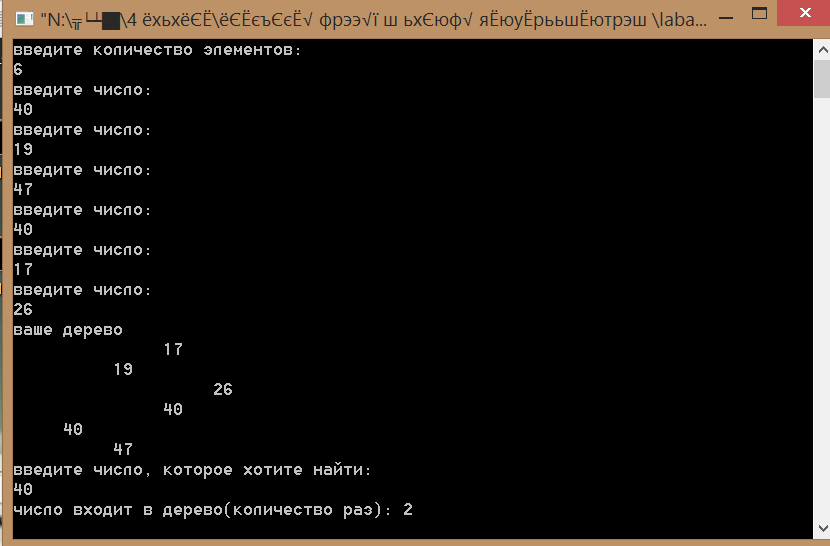
**4. Распечатки работы программы**

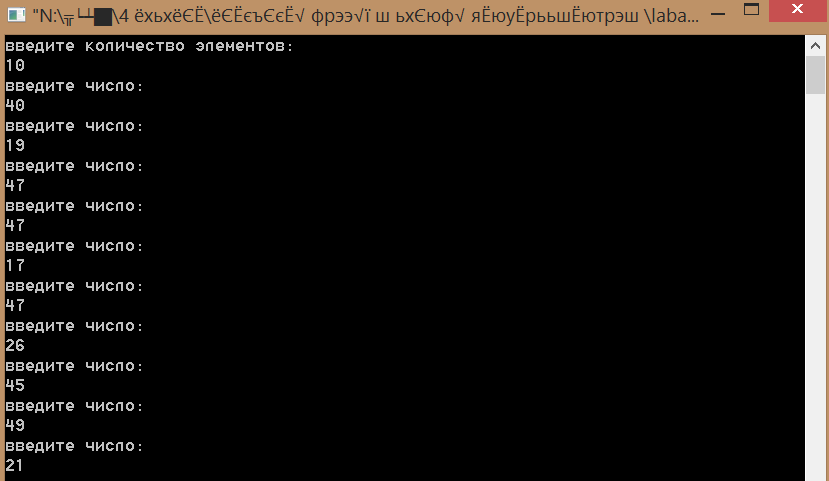
Ввод дерева

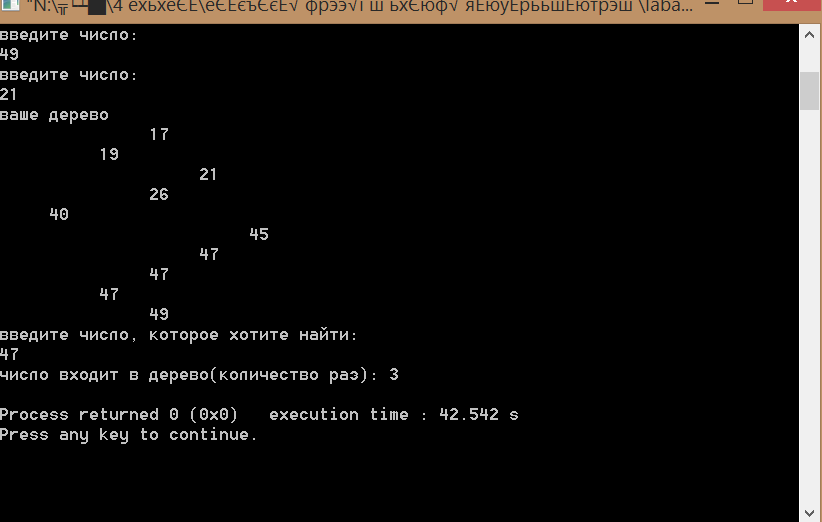


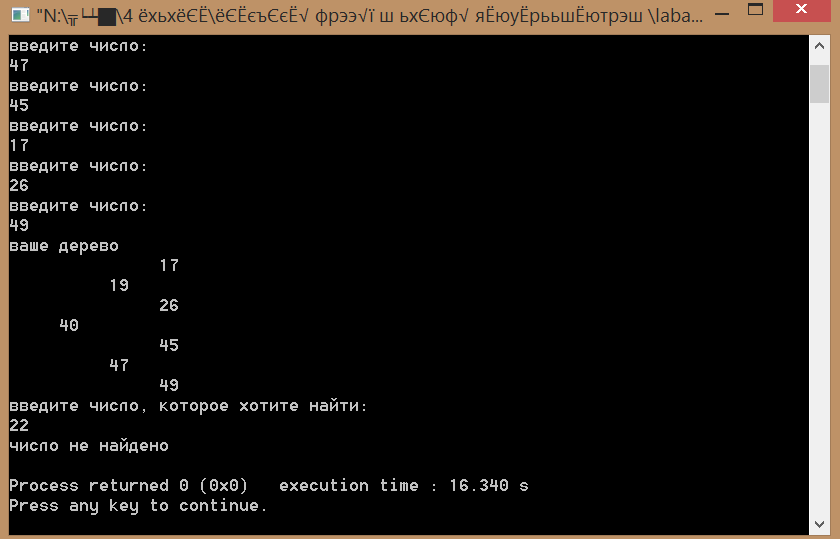
Поиск элемента в дереве











**5.Текст программы**

#include <iostream>

using namespace std;

struct node {

int info;

node \*l

node \*r;

};

node \*tree = NULL;

int res = 0;

// запись элемента в бинарное дерево

void push(int a, node \*\*t) {

if((\*t) == NULL) // Если дерева не существует

{

(\*t) = new node;

(\*t)->info = a;

(\*t)->l = (\*t)->r = NULL;

return;

}

// Дерево есть

if (a > (\*t)->info)

push(a, &(\*t)->r);

else

push(a, &(\*t)->l);

}

// вывод дерева

void print(node \*t, int u) {

if (t == NULL)

return;

else

{

// Иначе

print(t->l, ++u);

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

{

cout << " ";

}

cout << t->info << endl;

u--;

print(t->r,++u);

}

}

int find1(node \*t,int d)

{

if(t==NULL)

{

return res;

}

if(t->info == d)

{

res = res + 1;

}

if(d <= t->info)

{

// left

if(t->l != NULL)

{

find1(t->l,d);

}

else

{

return res;

}

}

else

{

//right

if(t->r)

{

find1(t->r,d);

}

else

{

return res;

}

}

}

int main() {

setlocale(LC\_ALL, "Russian");

int k;

int n; // Количество элементов

int s;

int fin;

res = 0;

cout << "введите количество элементов: \n";

cin >> n; // Вводим количество элементов

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

{

cout << "введите число: \n";

cin >> s; // Считываем элемент за элементом

push(s,&tree); // И каждый кладем в дерево

}

cout << "ваше дерево\n";

print(tree,0);

cout << "введите число, которое хотите найти: " << endl;

cin >> fin;

k = find1(tree,fin);

if(k == 0)

{

cout << "число не найдено" << endl;

}

else

cout << "число входит в дерево(количество раз): " << k << endl;

return 0;

}