**Лабоpатоpная pабота №2**

**Бинарные деревья**

Написать функцию формирования бинарного дерева, состоящего из целых чисел. Для представления дерева использовать динамические структуры данных. Количество элементов дерева, а также его вид задаются случайным образом. Произвести вывод элементов дерева тремя видами обхода. Используя информацию о выведенном дереве изобразить структуру одного из построенных деревьев в отчете. Выполнить по вариантам следующие задания:

1. Написать рекурсивную функцию, которая определяет, входит ли элемент X в дерево.

**Код программы.**

#include <iostream>

#include <time.h>

using namespace std;

struct Node {

int data;

Node\* left = NULL, \* right = NULL;

};

void add(Node\*& leaf) {

if (leaf == NULL) {

leaf = new Node;

leaf->data = rand() % 10;

}

else {

if (rand() % 2 == 0) {

add(leaf->right);

}

else {

add(leaf->left);

}

}

}

void Pryamoy(Node\*& leaf) {

if (leaf != NULL) {

cout << leaf->data << " ";

Pryamoy(leaf->left);

Pryamoy(leaf->right);

}

}

void Obratniy(Node\*& leaf) {

if (leaf != NULL) {

Obratniy(leaf->left);

Obratniy(leaf->right);

cout << leaf->data << " ";

}

}

void Simmetry(Node\*& leaf) {

if (leaf != NULL) {

Simmetry(leaf->left);

cout << leaf->data << " ";

Simmetry(leaf->right);

}

}

int find(Node\*& leaf, int x,int count) {

if (leaf != NULL) {

if (leaf->data == x) {

count += 1;

}

count = find(leaf->left, x, count);

count = find(leaf->right, x, count);

}

return count;

}

int main() {

setlocale(LC\_ALL, "Russian");

srand(time(0));

Node\* tree = NULL;

for (int i = 0; i < (rand() % 10 + 10); i++) {

add(tree);

}

cout << "Прямой:" << endl;

Pryamoy(tree);

cout << endl << "Обратный:" << endl;

Obratniy(tree);

cout << endl << "Симметричный:" << endl;

Simmetry(tree);

int x;

cout << "\nВведите искомый элемент в дереве:\n";

cin >> x;

int count = 0;

if (find(tree, x, count)) {

cout <<"Элемент " << x << " найден в дереве\n";

}

else {

cout << "Элемент " << x << " не найден в дереве\n";

}

system("pause");

return 0;

}

**Алгоритм**

Описываем структуру Node. Также описываем несколько методов для работы с нашим деревом:

add(добавляет элемент в дерево, случайно выбирает ветвь и случайно выбирает само число)

Pryamoy(печатает дерево на экран прямым обходом)

Obratniy(печатает дерево на экран обратным обходом)

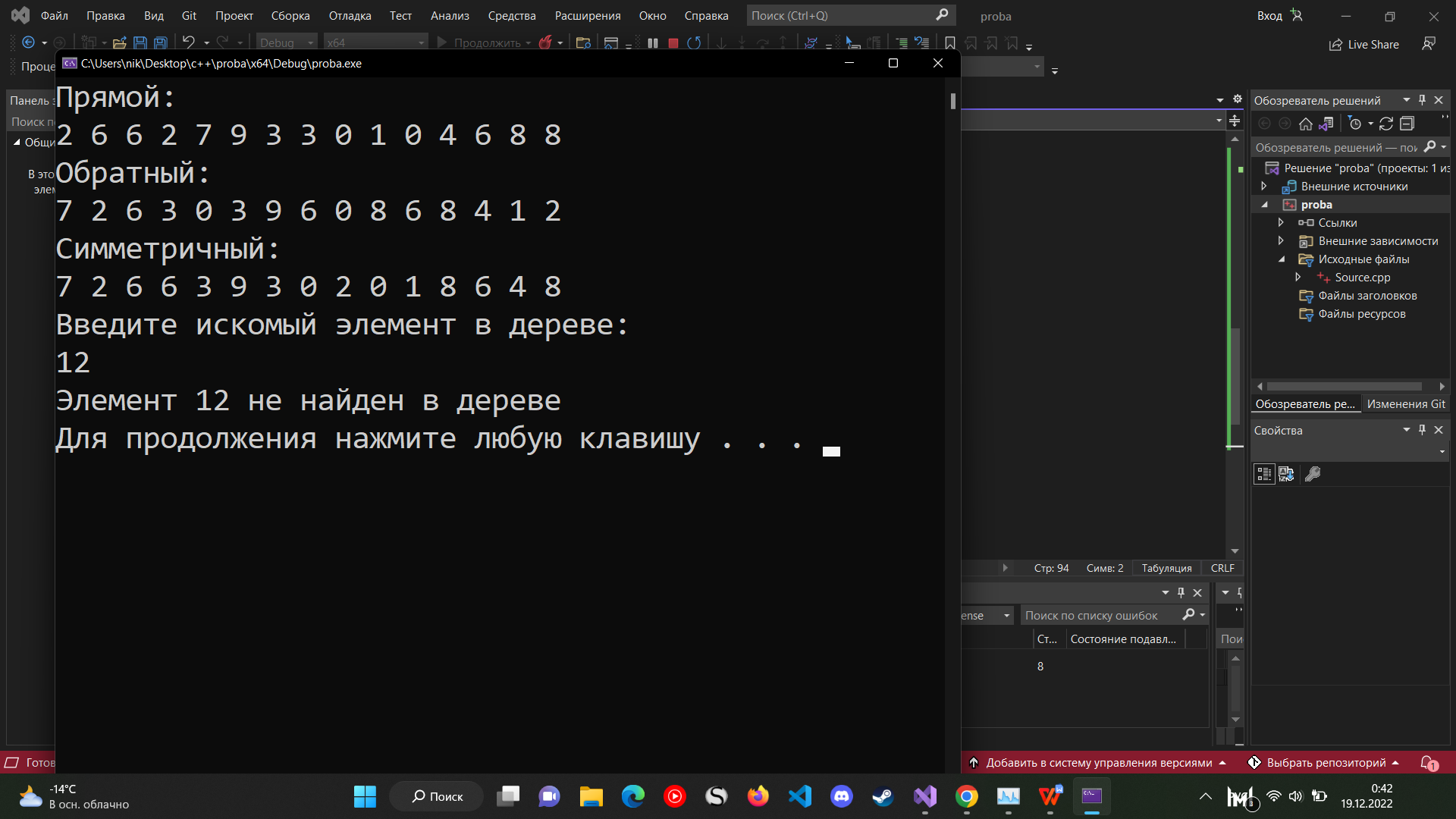
Simmetry(печатает дерево на экран симметричным обходом)

find(считает количество нахождения введеного числа в дереве)

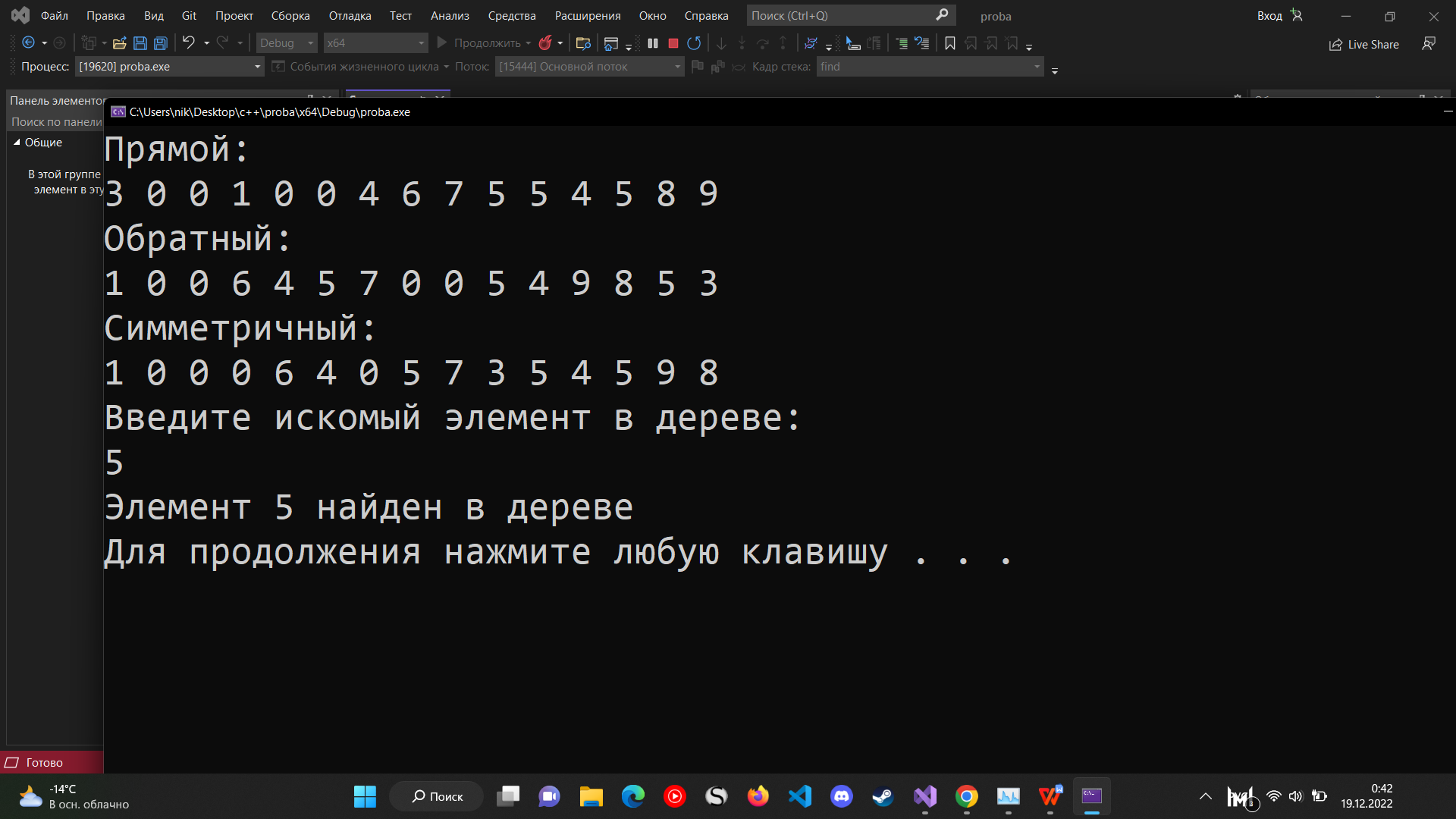
* Создаём экземпляр структуры Node tree.
* Добавляем в дерево от 10 до 19 листьев (число генерируется случайно) с помощью метода add и цикла for.
* Выводим дерево на экран с помощью методов Pryamoy, Obratniy и Simmetry. Тремя разными обходами (Прямым, обратным и симметричным соответственно).
* Вводим искомое число x.
* Объявляем переменную(count) для подсчёта количества нахождений числа x в нашем дереве.
* Если find(tree,x,count) > 0 - Выводим “Элемент x найден в дереве”, иначе - Вывод “Элемент x не найден в дереве”

**Тесты**

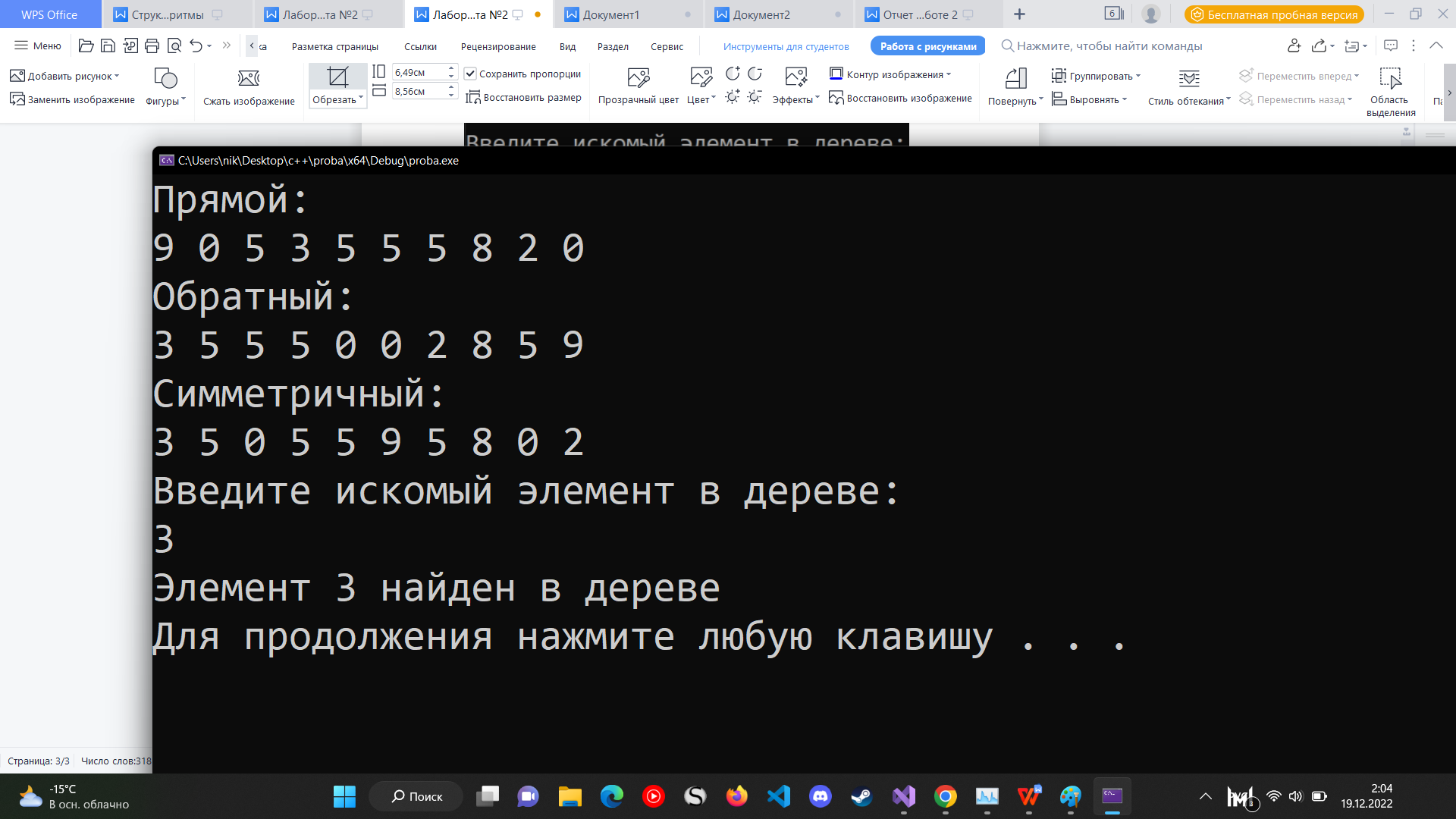
Тест 1.



Тест 2.



Тест 3



Пример графа по Тесту 3.

