Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 7

По дисциплине «Основы алгоритмизации и программирования»

На тему «Отладка программ»

Выполнил:

Студент 1 курса 9 группы

Павлович Ян Андреевич

Преподаватель: асс. Андронова М.В.

2023, Минск

1. Выполнить прокрутку для одной из программ лабораторной работы № 6.

#include <iostream>

//подключение библиотеки iostream для работы с вводом/выводом

**using** **namespace** std;

//объявляем программе, что мы работаем с пространством имен std

**void** main() {

//добавление функции

setlocale(LC\_CTYPE, "Russian");

    //добавляем возможность вывода текста на русском языке

**int** m = 16; //объявляем переменные типа integer и присваиваем m значение

**float** a = -1.4, w, r, j = 1.8; //добавляем переменные типа float и присваиваем a и j значения

**while** (j < 3.1) {

    //добавляем оператор while

w = tan(a / 3) + exp(a / m);

r = 0.9 \* sqrt(w + j) + abs(pow(a, 2) - 1);

        //производим вычисления

printf("Результаты вычислений:\n"); //вывод информации

printf("w = %2.3f\n", w); //вывод информации

printf("r = %2.3f\n\n", r); //вывод информации

j = j + 0.2; //добавляем к переменной j 0.1

}

}

Запись прокрутки для данного примера:

m = 16; a = -1.4; j = 1.8;

1.8 <= 3; w = tan(a / 3) + exp(a / m) = tan((-1.4) / 3) + exp((-1.4) / 16); r = 0.9 \* sqrt(w + j) + abs(pow(a, 2) - 1) = 0.9 \* sqrt(0.412 + 1.8) + abs(pow((-1.4), 2) - 1); вывод “Результаты вычислений:”; вывод w = 0.412 и r = 2.299;

j = j + 0.2 = 1.8 + 0.2 = 2;

2 <= 3; w = tan(a / 3) + exp(a / m) = tan((-1.4) / 3) + exp((-1.4) / 16); r = 0.9 \* sqrt(w + j) + abs(pow(a, 2) - 1) = 0.9 \* sqrt(0.412 + 2) + abs(pow((-1.4), 2) - 1); вывод “Результаты вычислений:”; вывод w = 0.412 и r = 2.358;

j = j + 0.2 = 2 + 0.2 = 2.2;

2.2 <= 3; w = tan(a / 3) + exp(a / m) = tan((-1.4) / 3) + exp((-1.4) / 16); r = 0.9 \* sqrt(w + j) + abs(pow(a, 2) - 1) = 0.9 \* sqrt(0.412 + 2.2) + abs(pow((-1.4), 2) - 1); вывод “Результаты вычислений:”; вывод w = 0.412 и r = 2.415;

j = j + 0.2 = 2.2 + 0.2 = 2.4;

2.4 <= 3; w = tan(a / 3) + exp(a / m) = tan((-1.4) / 3) + exp((-1.4) / 16); r = 0.9 \* sqrt(w + j) + abs(pow(a, 2) - 1) = 0.9 \* sqrt(0.412 + 2.4) + abs(pow((-1.4), 2) - 1); вывод “Результаты вычислений:”; вывод w = 0.412 и r = 2.469;

j = j + 0.2 = 2.4 + 0.2 = 2.6;

2.6 <= 3; w = tan(a / 3) + exp(a / m) = tan((-1.4) / 3) + exp((-1.4) / 16); r = 0.9 \* sqrt(w + j) + abs(pow(a, 2) - 1) = 0.9 \* sqrt(0.412 + 2.6) + abs(pow((-1.4), 2) - 1); вывод “Результаты вычислений:”; вывод w = 0.412 и r = 2.522;

j = j + 0.2 = 2.6 + 0.2 = 2.8;

2.8 <= 3; w = tan(a / 3) + exp(a / m) = tan((-1.4) / 3) + exp((-1.4) / 16); r = 0.9 \* sqrt(w + j) + abs(pow(a, 2) - 1) = 0.9 \* sqrt(0.412 + 2.8) + abs(pow((-1.4), 2) - 1); вывод “Результаты вычислений:”; вывод w = 0.412 и r = 2.573;

j = j + 0.2 = 2.8 + 0.2 = 3;

3 <= 3; w = tan(a / 3) + exp(a / m) = tan((-1.4) / 3) + exp((-1.4) / 16); r = 0.9 \* sqrt(w + j) + abs(pow(a, 2) - 1) = 0.9 \* sqrt(0.412 + 3) + abs(pow((-1.4), 2) - 1); вывод “Результаты вычислений:”; вывод w = 0.412 и r = 2.623;

j = j + 0.2 = 3 + 0.2 = 3.2;

3.2 > 3. Выполнение программы закончено.

1. В соответствии со своим вариантом написать программу и отладить ее для задачи, представленной в таблице ниже. Опробовать средства ***интерактивной*** отладки **Шаг с обходом**. Вызвать окно **Контрольное значение** **1**, поместить туда имена переменных и проконтролировать их значения в процессе выполнения программы.



#include <iostream>

//подключение библиотеки iostream для работы с вводом/выводом

**using** **namespace** std;

//объявляем программе, что мы работаем с пространством имен std

**void** main()

//добавление функции

{

**float** a = -4.2, d, f; //объявляем переменные типа float и присваиваем m значение

**int** t[5], i = 4, x = 0; //объявляем переменные типа integer и присваиваем m значение

t[0] = 5;

t[1] = 1;

t[2] = -3;

t[3] = 9;

t[4] = -1;

    //добавляе числа в массив

**for** (**int** y = 0; y < 5; y++) {

        //добавляем оператор for

d = i + 2 \* t[x] \* (1 + sqrt(3 \* pow(a, 2))); //производим вычисления

**if** (d >= t[x] \* (t[x] + i)) {

            //добавляем оператор if

f = t[x] \* i; //производим вычисления

}

**else** {

f = exp(t[x] - d) + 9; //производим вычисления

}

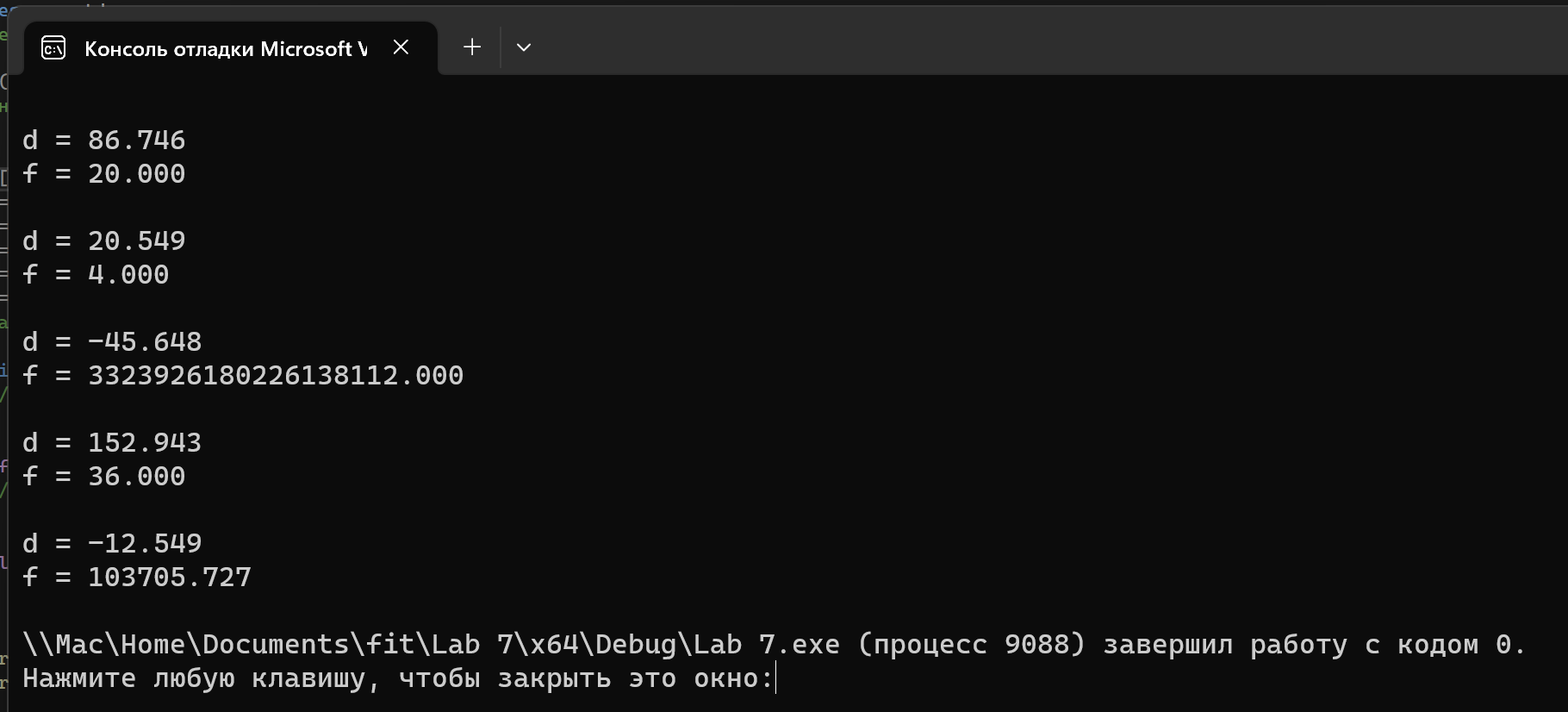
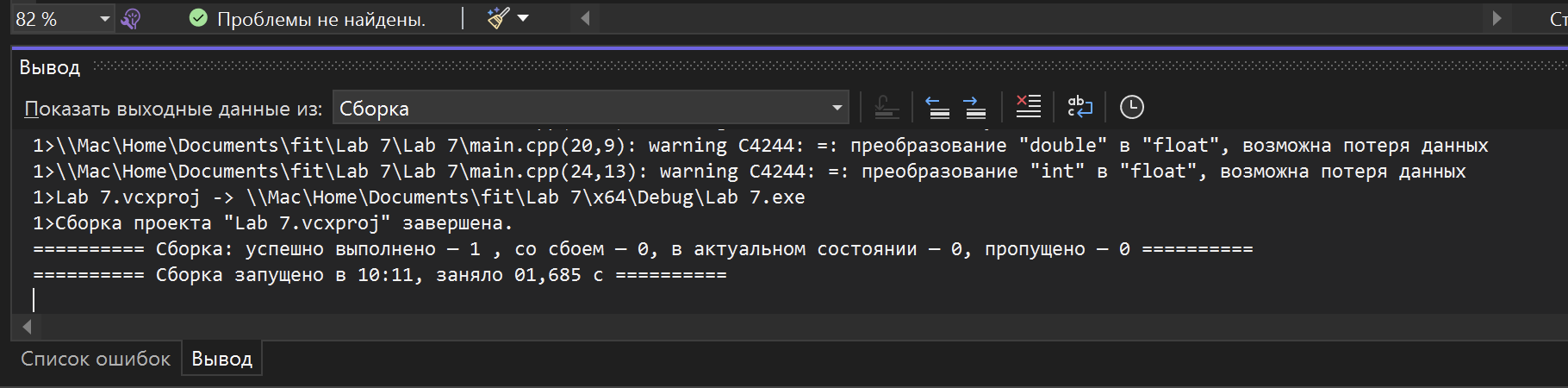
printf("\nd = %2.3f\n", d); //вывод информации

printf("f = %2.3f\n", f); //вывод информации

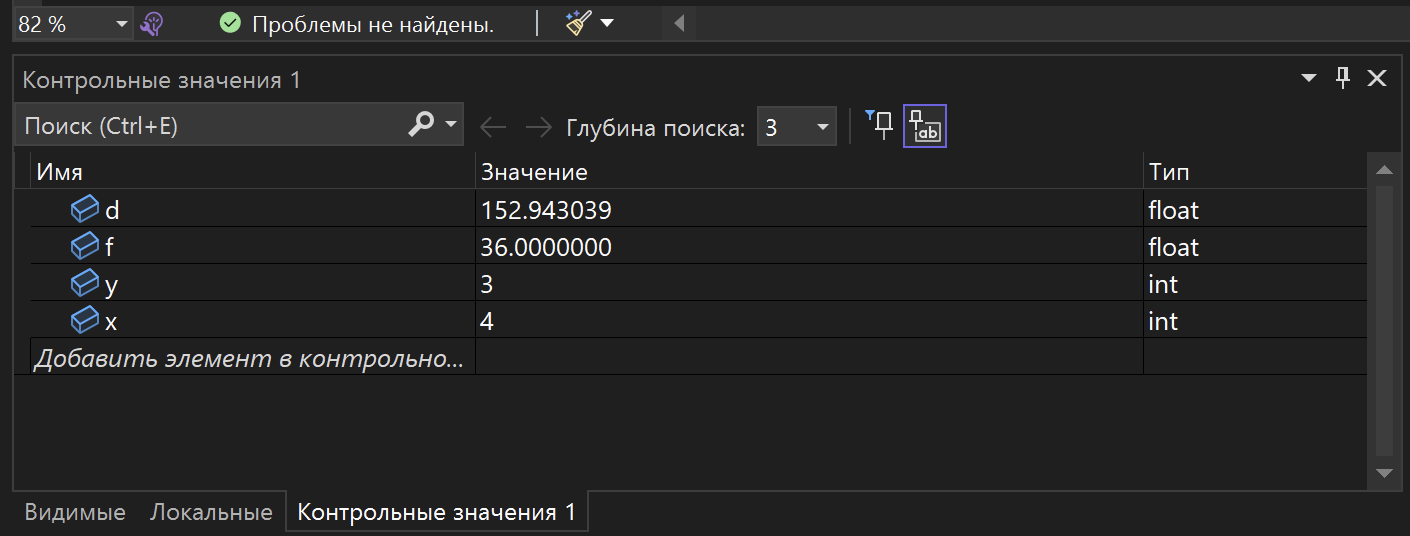
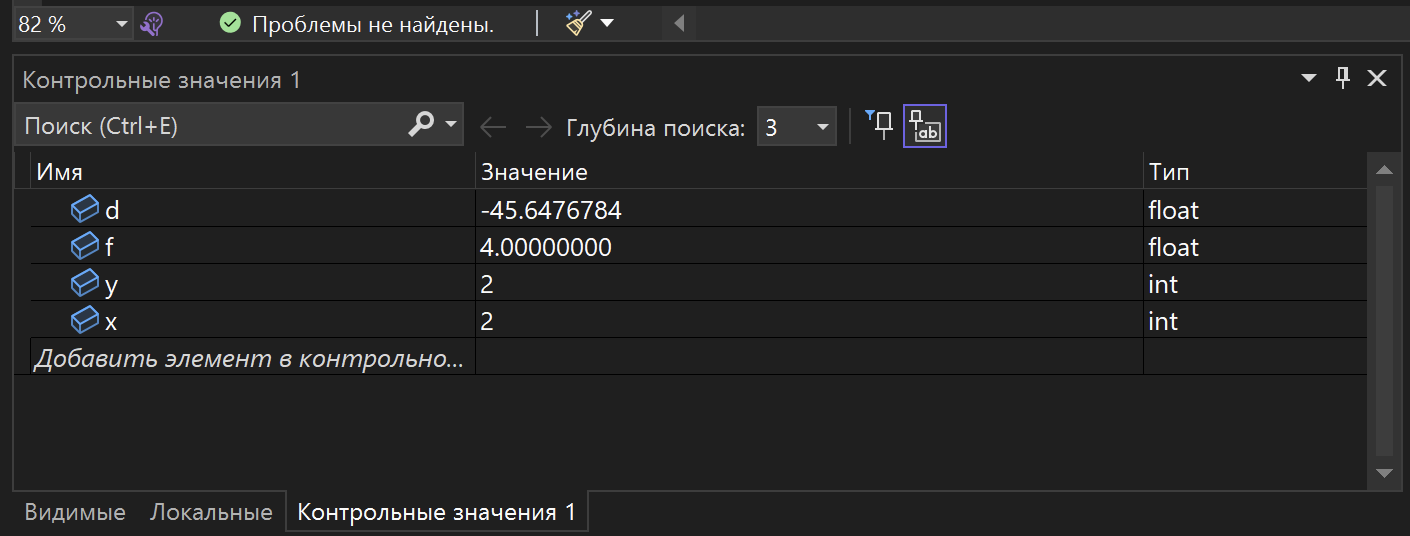
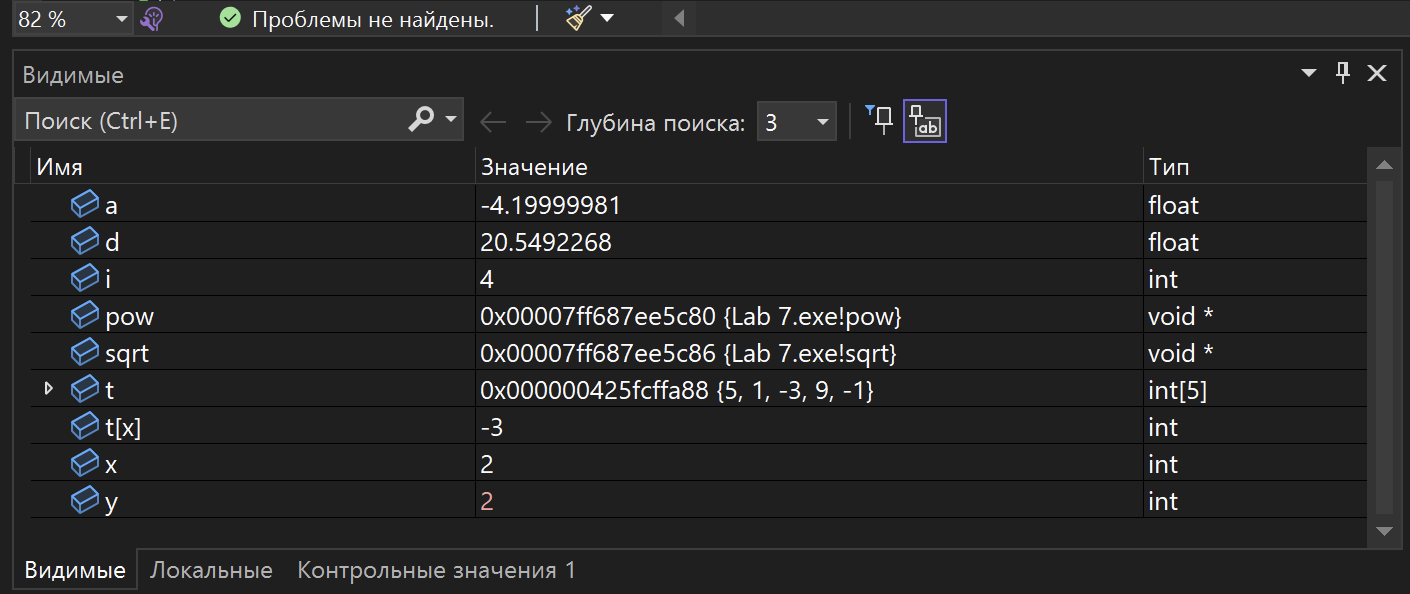
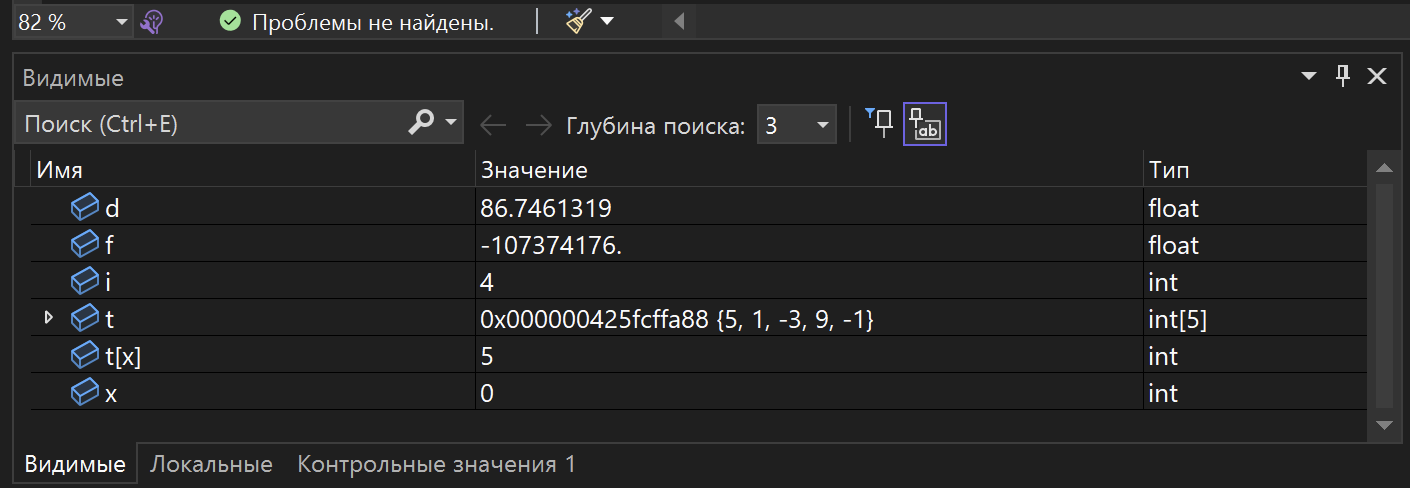
x++; //добавление к переменной x 1

}

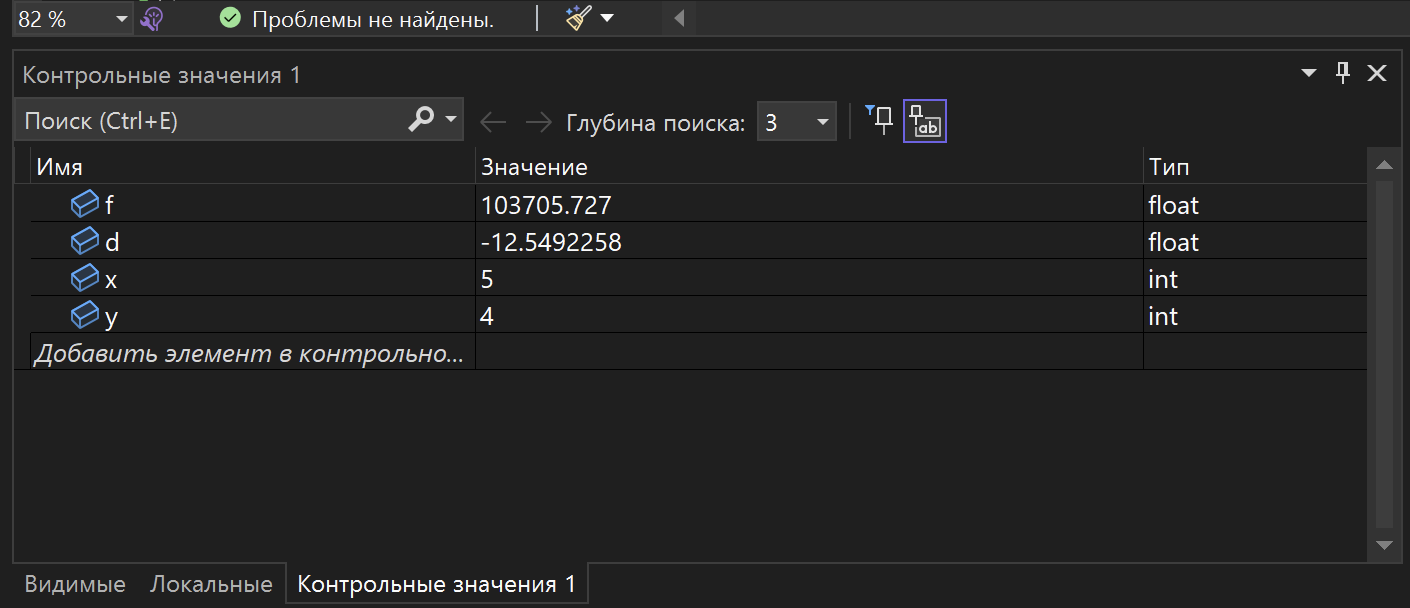
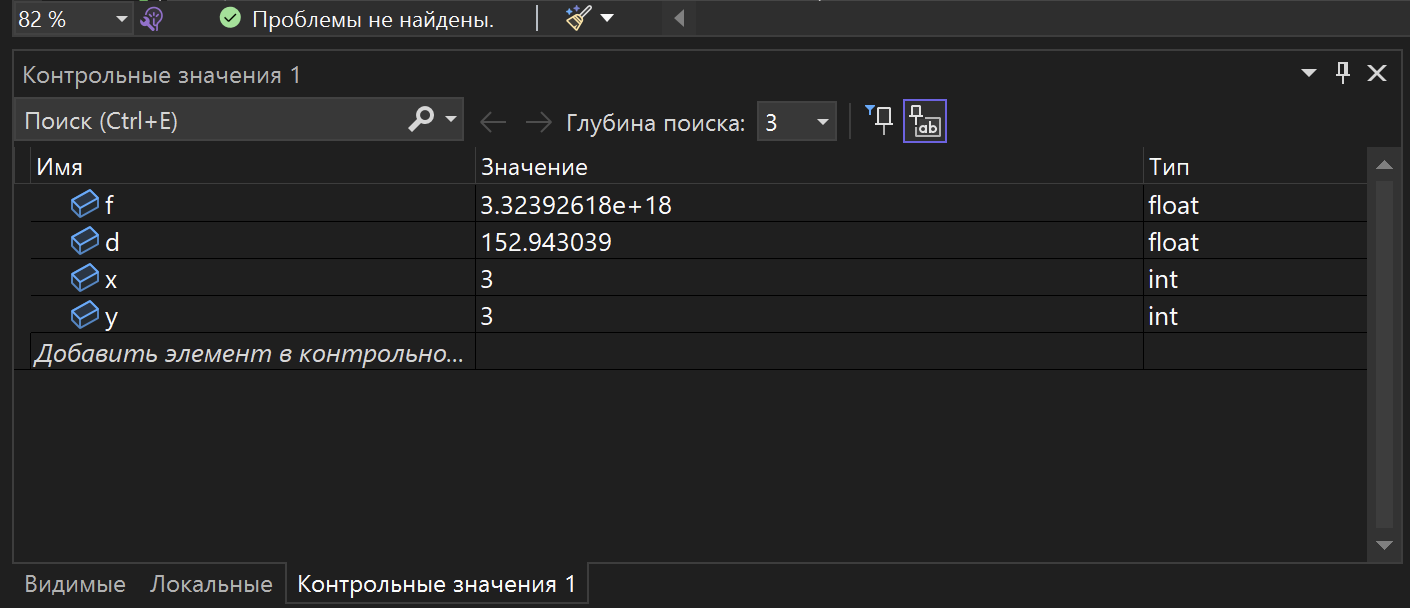
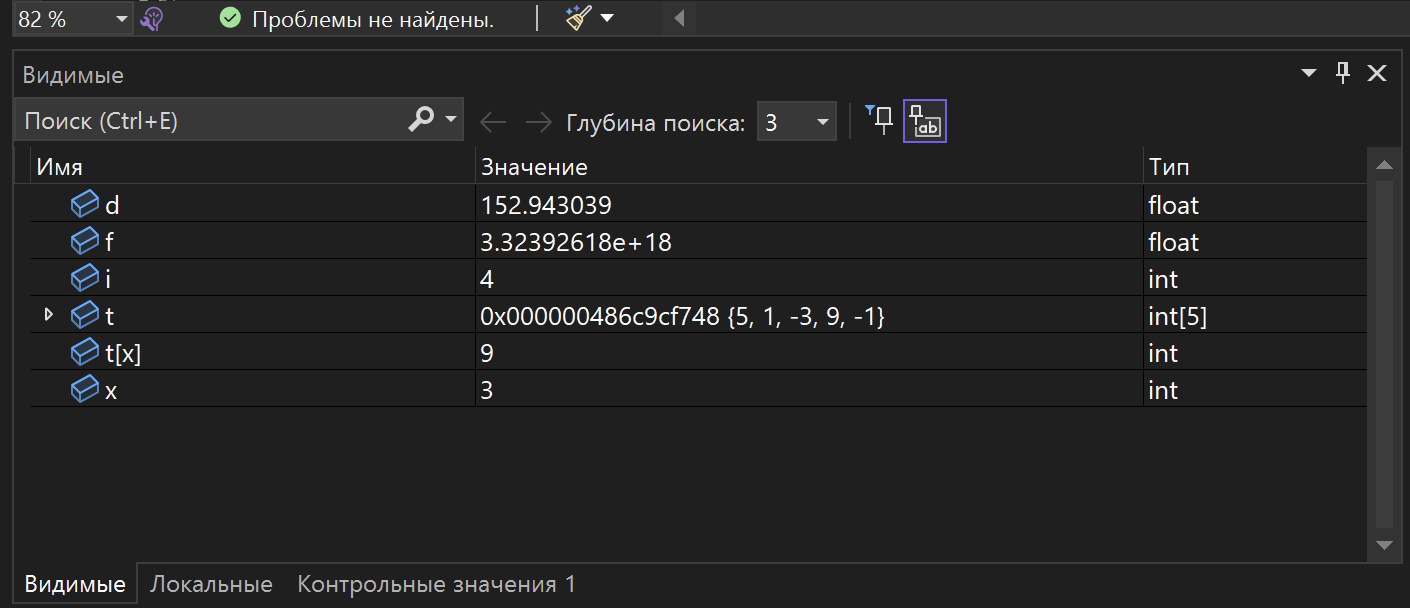
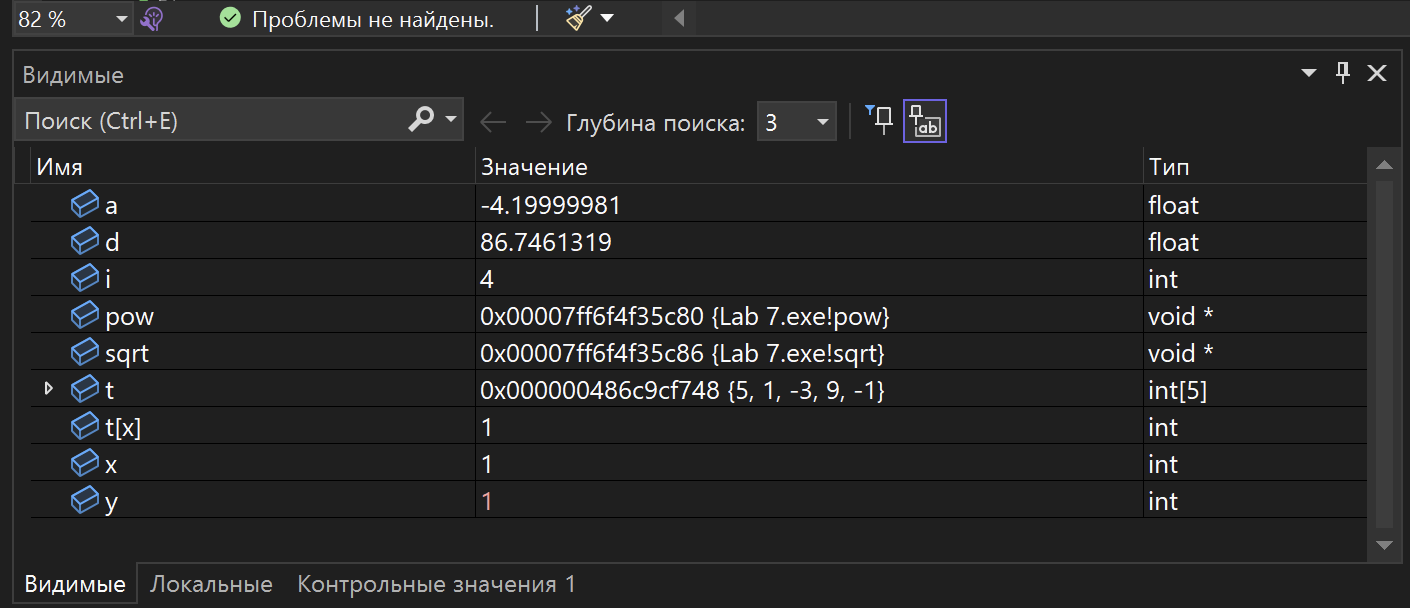
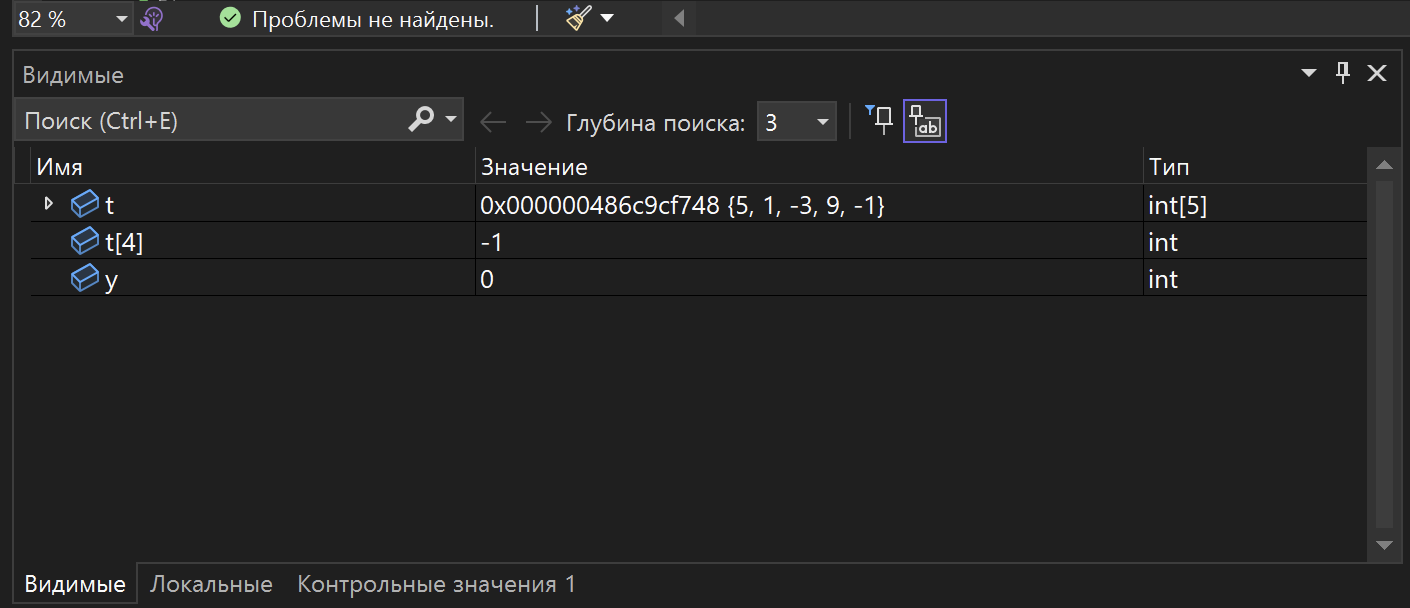
}



Шаг с обходом:



Шаг с заходом:



1. В соответствии со своим вариантом написать программу и отладить ее для задачи, представленной в таблице ниже. Опробовать средства ***планируемой*** отладки.

#include <iostream>

//подключение библиотеки iostream для работы с вводом/выводом

**using** **namespace** std;

//объявляем программе, что мы работаем с пространством имен std

**int** main() {

**int** f, d, b, count = 0; //объявляем переменные типа integer и присваиваем m значение

cout << "Enter f: "; //вывод информации

cin >> f; //ввод информации

**for** (**int** i = 10; i < 100; i++) {

        //добавляем оператор for

d = i / 10;  // получаем десятки (первую цифру)

b = i % 10; // получаем единицы (вторую цифру)

**if** (d + b == f) {

//добавляем оператор if

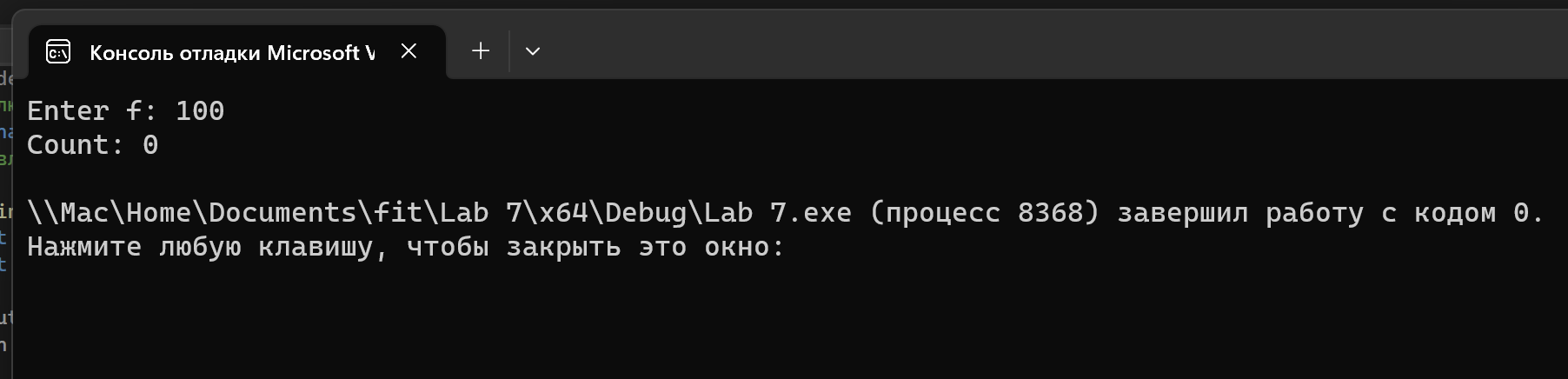
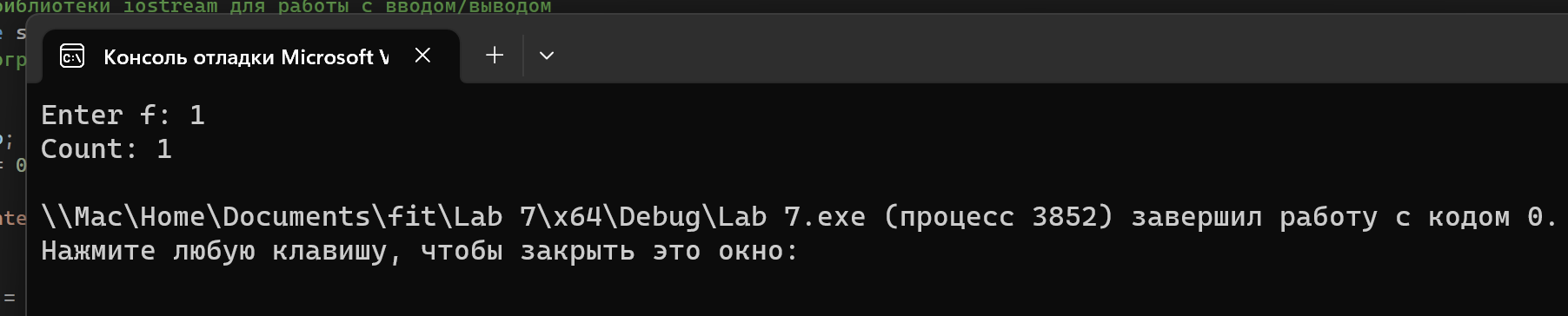
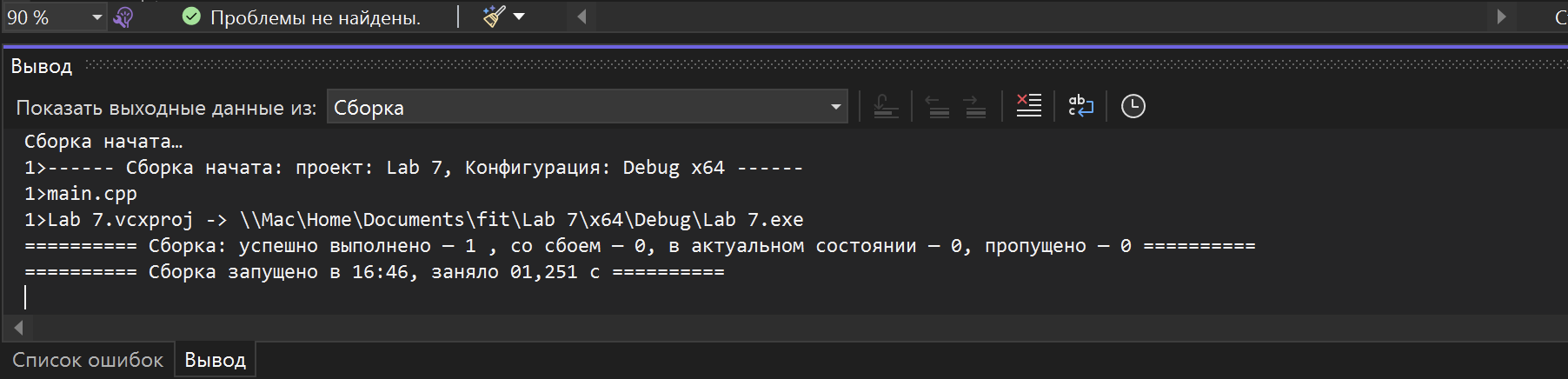
count++; //добавляем к count 1

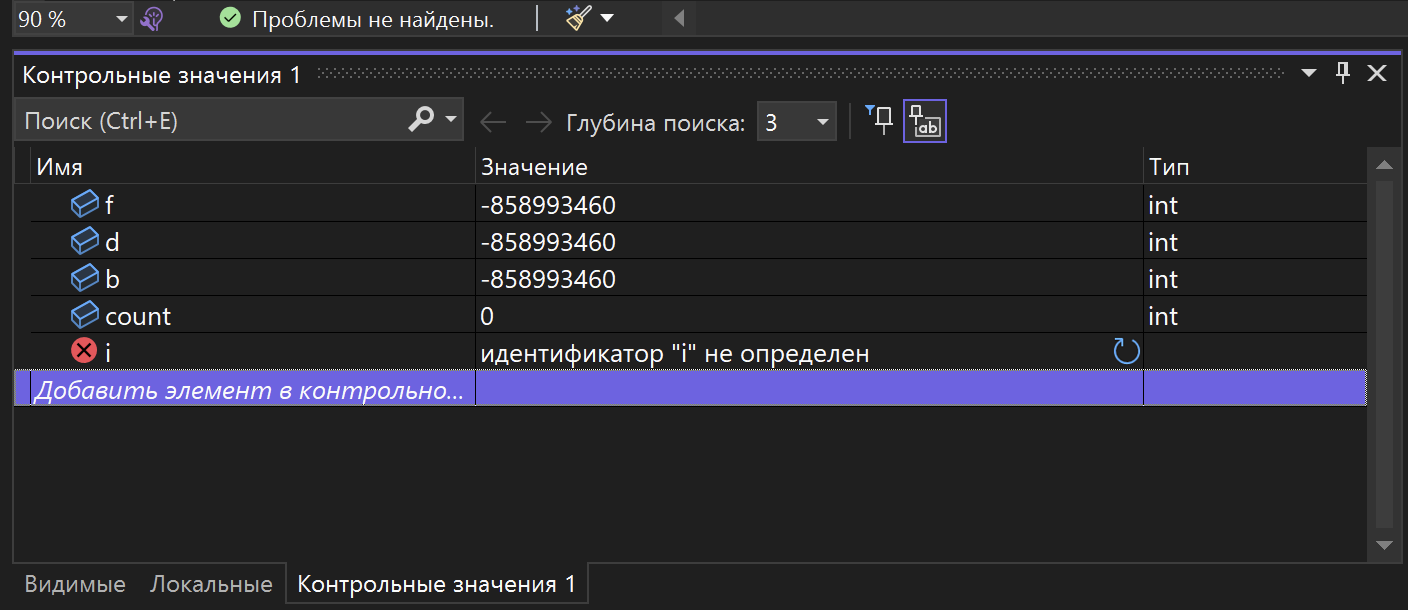
}

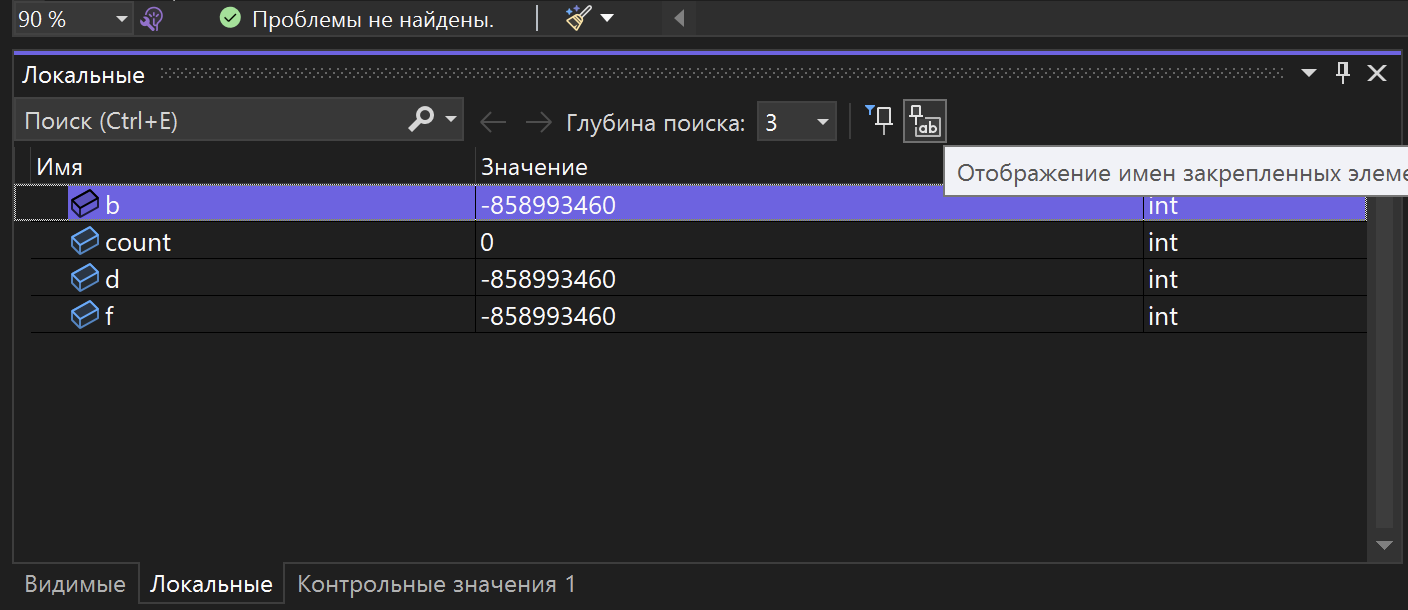
}

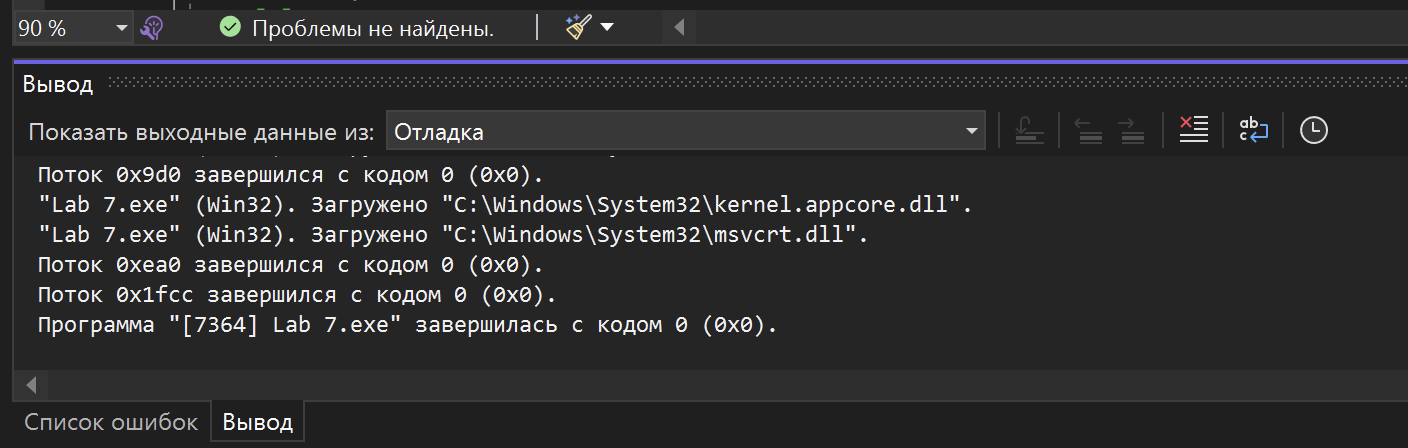
cout << "Count: " << count << endl; //вывод информации

}









Дополнительные задания:

1. Найти натуральное число, состоящее из трёх цифр, с возрастающими слева направо цифрами, являющееся полным квадратом. Число является полным квадратом, если квадратный корень из него – простое число (число 121 – полный квадрат, т. к. 121=11 ∙ 11, а 11 – простое число).

#include <iostream>

//подключение библиотеки iostream для работы с вводом/выводом

**using** **namespace** std;

//объявляем программе, что мы работаем с пространством имен std

**int** main() {

//добавляем функцию

setlocale(LC\_CTYPE, "Russian"); //добавляем возможность для работы с русским языком

**int** num = 100, a, b, c, d; //добавляем переменные типа integer и присваиваем значения

**while** (num < 1000) {

    //добавляем оператор while для проверки всех 3х значных чисел

d = num;

a = d % 10; //определяем последнюю цифру

d = d / 10;

b = d % 10; //определяем вторую цифру

d = d / 10;

c = d % 10; //определяем первую цифру

**if** (c < b && c < a && b < a) {

            //добалвляем оператор if для проверки числа на возрастание цифр

**if** (10 \* 10 == num) {

            //добавляем оператор if, а за ним и else if для проверки числа на полный квадрат

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

} **else** **if** (11 \* 11 == num) {

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

} **else** **if** (12 \* 12 == num) {

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

} **else** **if** (13 \* 13 == num) {

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

} **else** **if** (14 \* 14 == num) {

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

} **else** **if** (15 \* 15 == num) {

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

} **else** **if** (16 \* 16 == num) {

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

} **else** **if** (17 \* 17 == num) {

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

} **else** **if** (18 \* 18 == num) {

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

} **else** **if** (19 \* 19 == num) {

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

} **else** **if** (20 \* 20 == num) {

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

} **else** **if** (21 \* 21 == num) {

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

} **else** **if** (22 \* 22 == num) {

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

} **else** **if** (23 \* 23 == num) {

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

} **else** **if** (24 \* 24 == num) {

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

} **else** **if** (25 \* 25 == num) {

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

} **else** **if** (26 \* 26 == num) {

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

} **else** **if** (27 \* 27 == num) {

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

} **else** **if** (28 \* 28 == num) {

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

} **else** **if** (29 \* 29 == num) {

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

} **else** **if** (30 \* 30 == num) {

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

} **else** **if** (31 \* 31 == num) {

cout << "Число с возрастающими слева направо цифрами и является полным квадратом: " << num << endl;

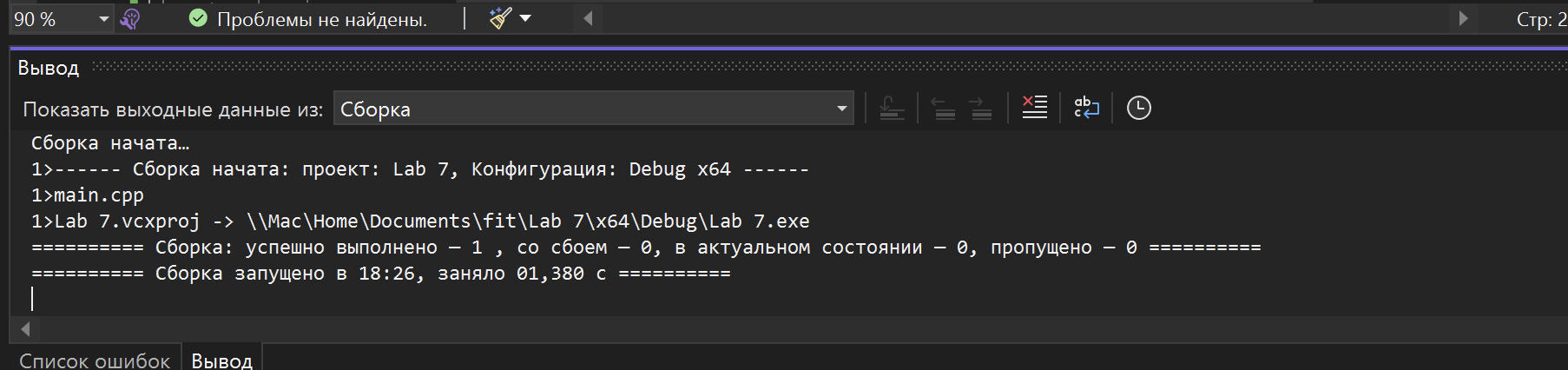
}

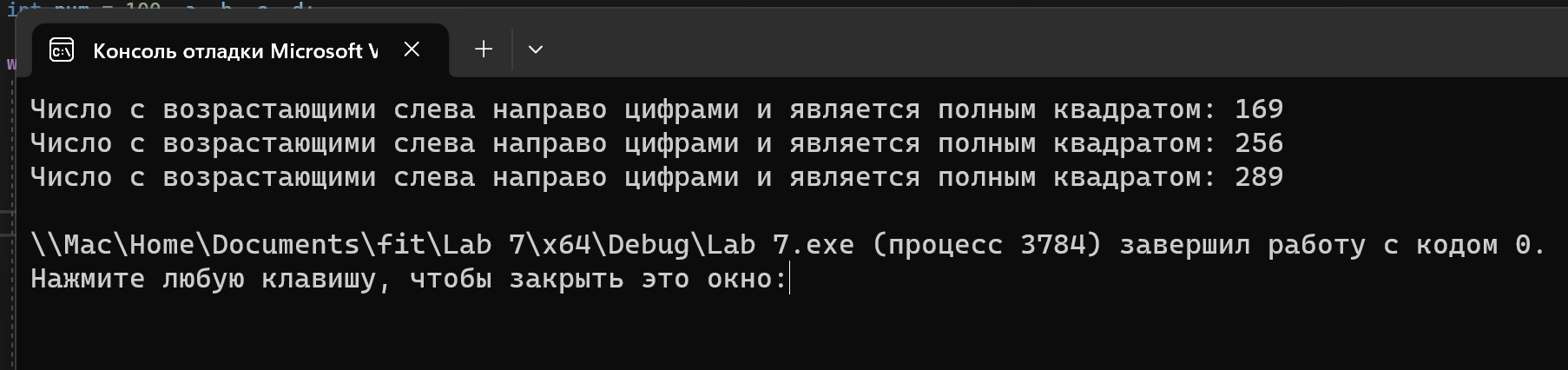
}

num++; //добавляем к num 1

}

}





3. Имеются два сосуда. В первом сосуде находится C1 литров воды, во втором – C2 литров воды. Из первого сосуда переливают половину воды во второй сосуд, затем из второго переливают половину в первый сосуд, и т. д. Сколько воды окажется в обоих сосудах после 12 переливаний?

#include <iostream>

//подключение библиотеки iostream для работы с вводом/выводом

**using** **namespace** std;

//объявляем программе, что мы работаем с пространством имен std

**int** main() {

    //добавляем функцию

setlocale(LC\_CTYPE, "Russian");

//добавляем возможность работы на русском языке

**float** C1, C2; //добавляем переменные типа float

cout << "Введите количетво литров в первом сосуде: "; //вывод информации

cin >> C1; //ввод

cout << "Введите количетво литров во втором сосуде: "; //вывод информации

cin >> C2; //ввод

**for** (**int** i = 0; i < 6; i++) { //добавляем оператор for

C2 + (C1 / 2); //добавляем в второй сосуд половину первого

C1 = C1 / 2; //убираем половину содержимого первого сосуда

C1 = C1 + (C2 / 2); //добавляем в первый сосуд половину второго

C2 = C2 / 2; //убираем половину содержимого второго сосуда

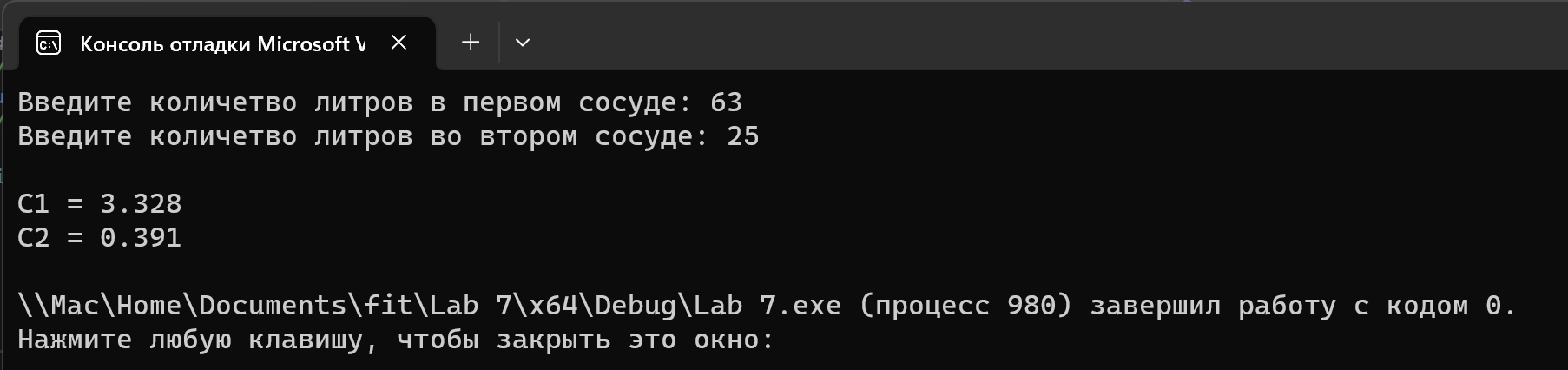
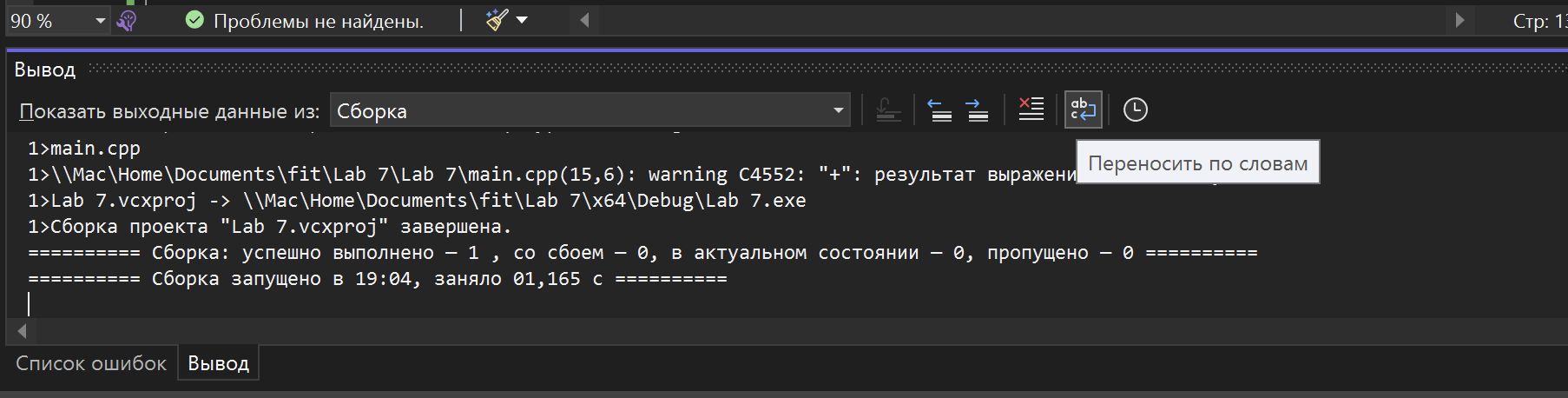
}

printf("\nC1 = %2.3f\n", C1);

printf("C2 = %2.3f\n", C2);

    //выводим количество литров в обоих сосудах после 12 переливаний

}



2. Составить алгоритм, определяющий, сколько существует способов набора одного рубля при помощи монет достоинством 50 коп., 20 коп., 5 коп. и 2 коп.

#include <iostream>

//подключение библиотеки iostream для работы с вводом/выводом

**int** main() {

    //добавляем функцию

setlocale(LC\_CTYPE, "Russian");//добавляем возможность вывода на русском языке

**int** result = 0; //добавляем переменню типа integer

**for** (**int** i = 0; i < 2; i++) {

        //добавляем оператор for для проверки монет 50коп, их нужно максимум 2

**for** (**int** j = 0; j < 5; j++) {

            //добавляем оператор for для проверки монет 20коп, их нужно максимум 5

**for** (**int** k = 0; k < 20; k++) {

                //добавляем оператор for для проверки монет 5коп, их нужно максимум 20

**for** (**int** l = 0; l < 50; l++) {

                    //добавляем оператор for для проверки монет 2коп, их нужно максимум 50

**if** (((i \* 50) + (j \* 20) + (k \* 5) + (l \* 2)) == 100) { //проверяем сумму монет

result++; //добавляем к результату 1

}

}

}

}

}

std::cout << "Количество способов набора одного рубля: " << result; //выводим количество решений

}

