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 7 группы

Ленкевич Павел Андреевич

Преподаватель: асс. Андронова М.В.

2023, Минск

**Вариант 9**

**Задание 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.

В отчете представить результаты в окне **Отладчика** (окно **Видимые** иокно **Контрольное значение** **1**).

Для той же программы представить результаты в окнах **Отладчика** для интерактивной отладки **Шаг с заходом**.

| **№** | **Формулы для вычислений** | **Исходные данные** |
| --- | --- | --- |
| 9 |  |  |

#include <iostream> //Подключение библиотеки

using namespace std; // Позволяет не писать std перед потоковыми операторами Ввода-вывода

void main() // Тело программы

{

setlocale(LC\_CTYPE, "Russian"); // Подключает русский язык

float c, d, f = -125 \* pow(10, -6), y = 1.7, t = 1; // Задаём значения переменных

while (t < 2.1) { // Цикл с проверкой условия в ()

c = 2 \* sin(f / 2) + log(t); // Расчёт значения переменной c

d = c >= 3 ? y \* exp(-2 \* t) + f : y - pow(y, 3); // Тернарный оператор: Если c>=3, то d= y \* exp(-2 \* t) + f, иначе d= y - pow(y, 3)

cout << "При t= " << t << "\t"; // Вывод значений переменных

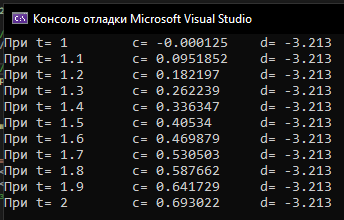
cout << "c= " << c << "\t";

cout << "d= " << d << endl;

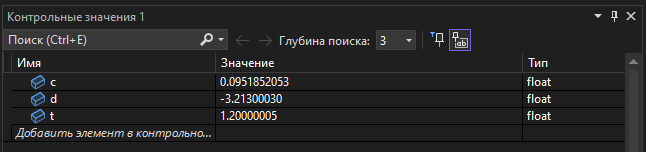
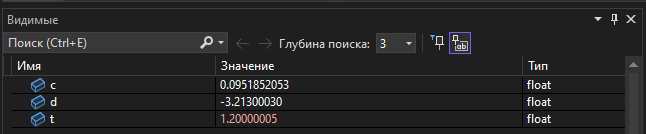
t += 0.1; // Изменение переменной для предотвращения бесконечности цикла и выполнения условия поставленного задачей

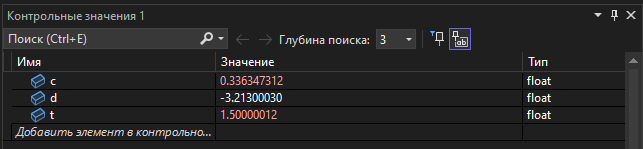
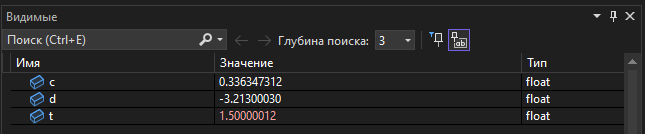
}

}

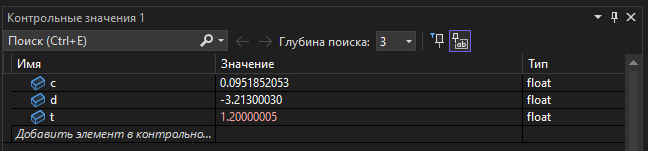
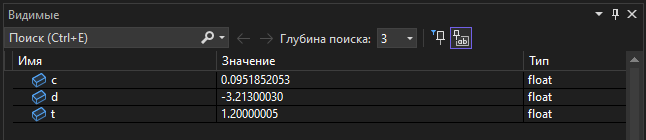
****

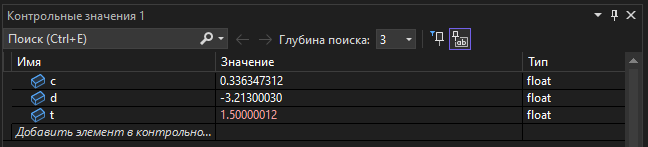
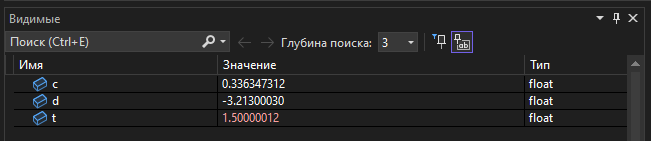
**Шаг с обходом**





**Шаг с заходом**

****

****

**Задание 2.**

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

В отчете представить результаты в окне **Отладчика** (окно **Локальные** и окно **Контрольное значение 1**).

Создать ***исполняемый файл*** и ознакомиться с файлами в папках проекта.

|  |  |
| --- | --- |
| 9 | Вывести на экран в возрастающем порядке все четырехзначные числа, в десятичной записи которых нет одинаковых цифр. |

#include <iostream> //Подключение библиотеки

using namespace std; // Позволяет не писать std перед потоковыми операторами Ввода-вывода

int main() // Тело программы

{

for (int a = 1; a < 10; a++) // Тело цикла с количеством повторов: в () задана целая переменная и её значение, условие выполнения и оператор инкремента

for (int b = 0; b < 10; b++)

{

if (a == b) { // Оператор условия, если a=b, то переходит к оператору continue

continue; // Оператор позволяющий пропустить цикл сразу перейдя к оператору инкремента b++

}

for (int c = 0; c < 10; c++)

{

if (c == a || c == b) { // Если c=a, или c=b, то цикл перейдёт сразу к c++

continue;

}

for (int d = 0; d < 10; d++)

if (a != b && a != c && a != d && b != c && b != d && c != d) // Проверка условия, чтоб исключить возможность повтора цифр

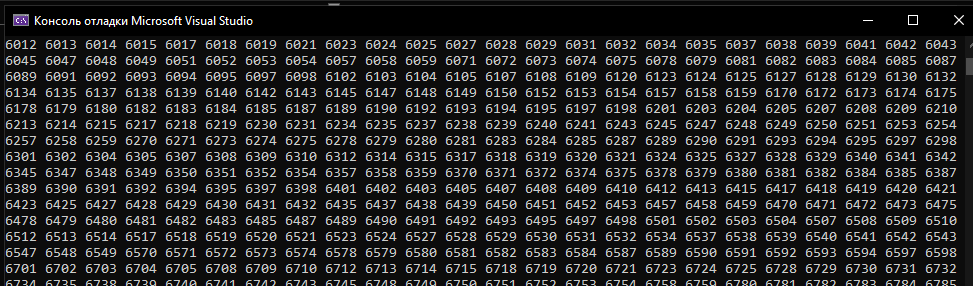
cout << a \* 1000 + b \* 100 + c \* 10 + d << " "; // Если условие выше верно, то вывод значения суммы переменных. Выводя их таким образом четырёхзначное число примет вид abcd, где d-единицы, c-десятки, b-сотни, a-тысячи

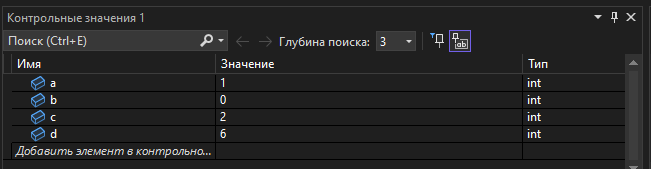
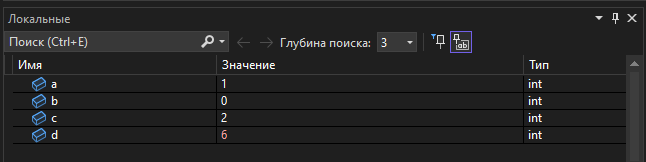
}

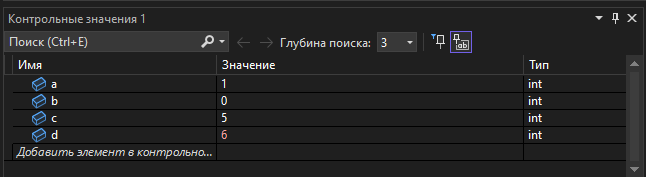
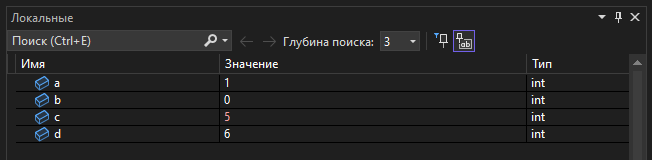
}

return 0;

}







**Дополнительные задания**

**Задание 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

using namespace std; // Позволяет не писать std перед потоковыми операторами Ввода-вывода

int main() // Тело программы

{

setlocale(LC\_CTYPE, "Russian"); // Подключает русский язык

for (int a = 13; a < 32; a++) { // Цикл с проверкой условия. С каждым повтором к переменной а прибовляется 1. В условии задана новая переменная

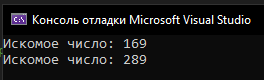
int n = a \* a; // Указание новой переменной и присвоение ей значения

if (n / 100 < n % 100 / 10 && n % 100 / 10 < n % 10 && a%2!=0 && a%3!=0 && a % 4 != 0 && a % 5 != 0) // Проверка условия: Если число n удовлетворяет задаче и a - простое число

cout << "Искомое число: " << n << endl; // То выводятся значение переменной

}

}



**Задание 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

using namespace std; // Позволяет не писать std перед потоковыми операторами Ввода-вывода

int main() // Тело программы

{

setlocale(LC\_CTYPE, "Russian"); // Подключает русский язык

int way = 0, sum;

for (int kop50 = 0; kop50 <3; kop50++) { // Цикл с условием, которое будет выполнятся пока монет по 50коп. < 3

for (int kop20 = 0; kop20 < 6; kop20++) { // Вложенный цикл с условием, которое выполняется пока монет по 20коп.<6

for (int kop5 = 0; kop5 < 21; kop5++) { // Вложенный цикл, условие выполняется пока монет по 5коп.<21

for (int kop2 = 0; kop2 < 51; kop2++) { // Вложенный цикл, условие выполняется пока монет по 2коп.<51

sum = (kop50 \* 50) + (kop20 \* 20) + (kop5\* 5) + (kop2 \* 2); // Присвоение значения переменной sum

if (sum == 100) { // Если sum=100 т.е. одному рублю, то к way прибавляется 1

way = way + 1;

}

}

}

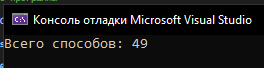
}

}

cout << "Всего способов: " << way << endl; // Вывод значения переменной

return 0; // Обнуляет переменные для корректного последующего запуска

}

****

**Задание 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

using namespace std; // Позволяет не писать std перед потоковыми операторами Ввода-вывода

float main() // Тело программы

{

setlocale(LC\_CTYPE, "Russian"); // Подключает русский язык

float c1, c2; // Добавляем переменные

cout << "Литров воды в первом сосуде: "; cin >> c1; // Присваиваем значения переменных

cout << "Литров воды в втором кувшине: "; cin >> c2;

for (int i = 0; i < 12; i++) // Цикл с условием, который повторится 12 раз

{

c1 /= 2; // Присвоение значений переменным в результате переливаний

c2 += c1;

c2 /= 2;

c1 += c2;

}

cout << "После 12 переливаний в первом кувшине: " << c1 << " литров, а в втором кувшине " << c2 << " литров."; // Вывод значений переменных

}

