

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение  
высшего образования «Томский государственный университет систем  
управления и радиоэлектроники»

Факультет систем управления (ФСУ)

Кафедра автоматизированных систем управления (АСУ)

Основы языка Ассемблер. Команды условных и безусловных переходов.  
отчет по лабораторной работе №4 по дисциплине  
«Вычислительная техника»

Обучающийся гр. 431-3

\_\_\_\_\_ Сергиевский Д.В.

«\_\_» \_\_\_\_\_ 2022 г.

Проверил: доцент каф. АСУ, к.т.н.

\_\_\_\_\_ Алферов С.М.

«\_\_» \_\_\_\_\_ 2022 г.

Томск 2022

## Содержание

Введение.....	3
1 Ход работы.....	4
Вывод.....	5
Приложение А.....	6

## Введение

В рамках данной лабораторной работы необходимо написать небольшую программу на C++ с использованием ассемблерных вставок для закрепления теоретического материала по теме.

Задание представлено на Рисунке 1, вариант задания — на Рисунке 2.

### 3. Задание

Задание из двух частей. В первом задании необходимо подсчитать количество чисел, соответствующих определенному условию на некотором числовом промежутке. Во втором задании требуется модифицировать свою программу так, чтобы найти и вывести на экран двухсотое число или пару чисел соответствующих заданному условию. Если таких чисел меньше двухсот, то вывести об этом сообщение на экран. Для выполнения задания выделяется 8 часов аудиторного времени.

Рисунок 1 - задание

3) На промежутке от 1 до 100 000. Подсчитать количество таких чисел  $X$ , что  $(X+X-2)$  - делится на 3 без остатка. Ответ вывести на экран.

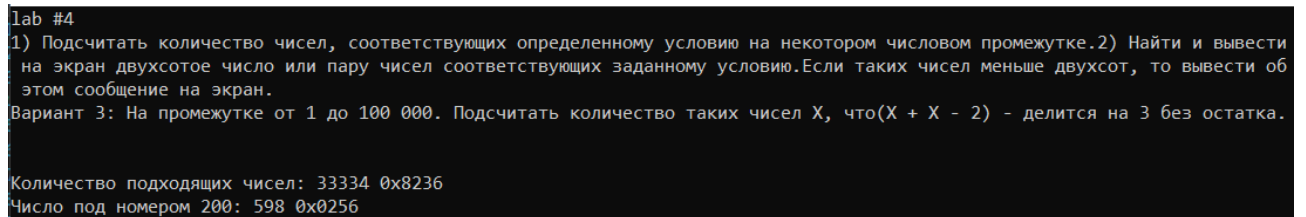
Рисунок 2 - вариант задания

## 1 Ход работы

В рамках данной лабораторной работы была написана небольшая программа, осуществляющая поиск и подсчёт чисел из промежутка, попадающих под указанное в задании условие.

Исходный код программы приведен в Приложении А.

Вывод программы представлен на Рисунке 1.1.



```
lab #4
1) Подсчитать количество чисел, соответствующих определенному условию на некотором числовом промежутке.2) Найти и вывести
на экран двухсотое число или пару чисел соответствующих заданному условию.Если таких чисел меньше двухсот, то вывести об
этом сообщение на экран.
Вариант 3: На промежутке от 1 до 100 000. Подсчитать количество таких чисел X, что  $(X + X - 2)$  - делится на 3 без остатка.

Количество подходящих чисел: 33334 0x8236
Число под номером 200: 598 0x0256
```

Рисунок 1.1 - вывод программы

## **Вывод**

В результате данной лабораторной работы теоретические знания по теме «Команды условных и безусловных переходов» были подкреплены практическим опытом.

## Приложение А

Файл s3\_cote\_lab4.cpp, точка входа и решение задачи.

Также код доступен по ссылке: [github.com/serdenvl/s3\\_cote\\_labs](https://github.com/serdenvl/s3_cote_labs)

```
// s3_cote_lab4.cpp : Этот файл содержит функцию "main". Здесь начинается и заканчивается
// выполнение программы.

#include <iostream>

using namespace std;

constexpr auto lab_num = 4;

constexpr auto description = "1) Подсчитать количество чисел, " \
"соответствующих определенному условию на некотором числовом промежутке." \
"2) Найти и вывести на экран двухсотое число или пару чисел соответствующих заданному \
условию." \
"Если таких чисел меньше двухсот, то вывести об этом сообщение на экран.\n" \
"Вариант 3: На промежутке от 1 до 100 000. " \
"Подсчитать количество таких чисел X, что  $(X + X - 2)$  - делится на 3 без остатка.\n";

int main()
{
    setlocale(LC_ALL, "Russian");

    cout << "lab #" << lab_num << endl;
```

```

cout << description << endl;

int number_of_matching_numbers = -1;
int number_under_number_200 = -1;

__asm {
    // EBX = 0 :счётчик
    mov EBX, 0

    // :temp
    push EBX

    // for: ECX = 1
    mov ECX, 1

    loop_start:

    // EAX = ECX
    mov EAX, ECX

    // [EDX, EAX] = EAX * 2
    mov EBX, 2
    imul EBX

    // [EDX, EAX] -= 2

```

```
sub EAX, 2

sbb EDX, 0 // assert EDX == 0,

// [EDX, EAX] %/= 3

mov EBX, 3

idiv EBX


// if EDX == 0

cmp EDX, 0

jne number_not_match


// :счётчик

pop EBX


// EBX += 1

inc EBX


// if EBX == 200

cmp EBX, 200

jne number_no_200


//

mov dword ptr number_under_number_200, ECX
```



```

number_no_200:

// : temp

push EBX

number_not_match:

// for: ECX <= 100000

cmp ECX, 100000

je loop_end

// for: ECX++

inc ECX

jmp loop_start

loop_end:

// : счётчик

pop EBX

mov dword ptr number_of_matching_numbers, EBX
}

printf("Количество подходящих чисел: %d 0x%04X\n", number_of_matching_numbers,
number_of_matching_numbers);

printf("Число под номером 200: ");

```

```
if (number_under_number_200 != -1)

    printf("%d 0x%04X\n", number_under_number_200, number_under_number_200);

else

    printf("чисел меньше 200\n");

}

// Запуск программы: CTRL+F5 или меню "Отладка" > "Запуск без отладки"

// Отладка программы: F5 или меню "Отладка" > "Запустить отладку"


// Советы по началу работы

// 1. В окне обозревателя решений можно добавлять файлы и управлять ими.

// 2. В окне Team Explorer можно подключиться к системе управления версиями.

// 3. В окне "Выходные данные" можно просматривать выходные данные сборки и другие сообщения.

// 4. В окне "Список ошибок" можно просматривать ошибки.

// 5. Последовательно выберите пункты меню "Проект" > "Добавить новый элемент", чтобы создать файлы кода, или "Проект" > "Добавить существующий элемент", чтобы добавить в проект существующие файлы кода.

// 6. Чтобы снова открыть этот проект позже, выберите пункты меню "Файл" > "Открыть" > "Проект" и выберите SLN-файл.
```