

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования

**«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)**

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»

Студент гр. 431-3
_____ Гурулёв А.В
«07» ноября 2022 г.

Руководитель
_____ Алфёров С.М.
«___» _____ 2022 г.

Томск 2022

Оглавление

ВВЕДЕНИЕ	3
1 РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ	4
ВЫВОД	5
Приложение А	6

ВВЕДЕНИЕ

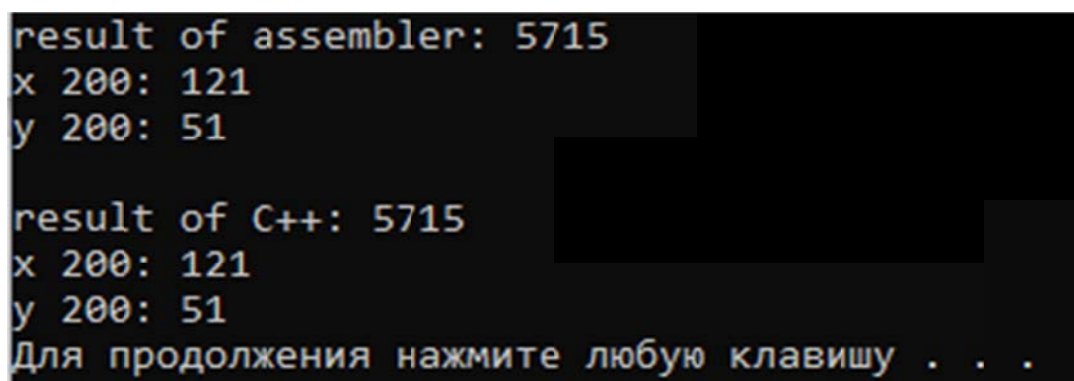
Цель работы – Получить навыки использования операторов сравнения (cmp), условных(Jxx), безусловных(JMP) переходов и циклов(LOOP).

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.
- 7) На промежутке от -128 до 127. Подсчитать количество таких пар чисел X и Y, что $50 < (X - Y) \leq 80$. Ответ вывести на экран

1 РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ

На рисунке 1.1 представлен результат работы программы.



```
result of assembler: 5715
x 200: 121
y 200: 51

result of C++: 5715
x 200: 121
y 200: 51
Для продолжения нажмите любую клавишу . . .
```

Рисунок 1.1 - Результат работы программы

ВЫВОД

Я получил навыки использования операторов сравнения (cmp), условных(Jxx), безусловных(JMP) переходов и циклов(LOOP).

Приложение А

Листинг кода

```
#include <iostream>
using namespace std;

//7 Вариант
int main()
{
    _int16 count = 0;
    _int16 x = 0;
    _int16 y = 0;

    _int16 xOut = 0;
    _int16 yOut = 0;

    _asm
    {
        //Счётчик для X
        mov cx, 256;

        //Метка возврата для X
        pointone:

        //Заносим в X значение счётчика(X = CX - 129)
        mov x, cx;

        //Находим X
        sub x, 129;

        //Переносим данные CX в стек тк далее идет перебор Y
        push cx;

        //Счётчик для Y
        mov cx, 256;

        //Метка возврата для Y
        pointtwo:

        //Заносим в Y значение счётчика(Y = CX - 129)
        mov y, cx;

        //Находим Y
        sub y, 129;

        //Находим x-y
        mov ax, x;
        sub ax, y;

        //Сравниваем результат с 50
        cmp ax, 50;

        //Если 50 < результат, то сравниваем дальше
        jle exittwo;

        //Сравниваем результат с 80
        cmp ax, 80;

        //Если результат <= 80, то увеличиваем count
        jg exitone;

        inc count;
```

```

        //Ищем 200ую пару
        cmp count, 200;
        jne exitThree;

        mov ax, x;
        mov xOut, ax;
        mov ax, y;
        mov yOut, ax;

        exitThree:

        //Иначе - выход
        exitone:
        exittwo:

        //"Цикл" для перебора значений Y
        loop pointtwo;

        //Забираем значения CX из стека, тк возвращаемся к перебору X
        pop cx;

        //"Цикл" для перебора значений X
        loop pointone;
    }

    cout << "result of assembler: " << count << "\nx 200: " << xOut << "\ny 200: " <<
    yOut << endl;

    //Блок на C++ для проверки предыдущего кода

    xOut = 0;
    yOut = 0;
    count = 0;

    for (int i = 256; i > 0; i--)
    {
        x = i - 129;
        for (int j = 256; j > 0; j--)
        {
            y = j - 129;
            if ((50 < (x-y)) && ((x-y) <= 80))
            {
                count++;
            }
            if (count == 200)
            {
                xOut = x;
                yOut = y;
            }
        }
    }

    cout << "\nresult of C++: " << count << "\nx 200: " << xOut << "\ny 200: " << yOut
    << endl;

    system("pause");

    return 0;
}

```