

Министерство науки и высшего образования Российской Федерации

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

Факультет систем управления (ФСУ)

Кафедра автоматизированных систем управления (АСУ)

Основы языка Ассемблер. Арифметические операции.  
отчет по лабораторной работе №3 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.

### **ЗАДАНИЕ**

Вычислить целочисленное выражение, указанное в варианте задания. При этом, и операнды и результаты вычислений следует выводить как в десятичном, так и в шестнадцатеричном виде.

Рисунок 1 - задание

3) A (byte), B (byte), C (dword).  
Вычислить:  $D = (C - A) / B$ ;  $E = (A * b) + B$ .

Рисунок 2 - вариант задания

## 1 Ход работы

В рамках данной лабораторной работы была написана небольшая программа, производящая арифметические вычисления над введенными переменными согласно указанной в задании формуле.

Исходный код программы приведен в Приложении А.

Вывод программы представлен на Рисунке 1.1.

```
lab #2
Вычислить целочисленное выражение, указанное в варианте задания.
При этом, и операнды и результаты вычислений следует выводить как в десятичном, так и в шестнадцатеричном виде.
3) A (byte), B (byte), C (dword).
Вычислить D = (C-A)/B; E = (A*B)+B

Input:
char A = 11
char B = 22
int C = 333
A = 11 0x0B
B = 22 0x16
C = 333 0x0000014D

Output:
D = (C-A)/B = 14 0x000E
E = (A*B)+B = 264 0x0108
```

Рисунок 1.1 - вывод программы

## **Вывод**

В результате данной лабораторной работы теоретические знания по теме «Арифметические операции» были подкреплены практическим опытом.

## Приложение А

Файл s3\_cote\_lab3.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_lab3.cpp : Этот файл содержит функцию "main". Здесь начинается и заканчивается
// выполнение программы.

#include <iostream>

using namespace std;

constexpr auto lab_num = 3;

constexpr auto description = "Вычислить целочисленное выражение, указанное в варианте
задания.\n" \
    "При этом, и операнды и результаты вычислений следует выводить как в
десятичном," \
    "так и в шестнадцатеричном виде.\n" \
    "3) A (byte), B (byte), C (dword).\n" \
    "Вычислить  $D = (C-A)/B$ ;  $E = (A*B)+B$ \n";

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

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

    cout << description << endl;
```

```

char A;

char B;

int C;

short D, E;


cout << "Input: " << endl;

cout << "char A = "; scanf_s("%hhd", &A);

cout << "char B = "; scanf_s("%hhd", &B);

cout << "int C = "; scanf_s("%d", &C);


printf("A = %d 0x%02X\n", A, A);

printf("B = %d 0x%02X\n", B, B);

printf("C = %d 0x%08X\n", C, C);


// D = (C-A)/B

// (int - char) / char = int / char = short


__asm {

    // [DX, AX] = int C

    mov AX, word ptr C + 0

    mov DX, word ptr C + 2


    // [DX, AX] -= char A

```

```

sub AL, byte ptr A

sbb AH, 0

sbb DX, 0


// BX = (short) char B

xchg AX, BX

mov AL, byte ptr B

cbw

xchg AX, BX


// [DX, AX] %/= BX

idiv BX


// short D = AX

mov word ptr D, AX
}


// E = (A*B)+B

// char * char + char = short + char = short


__asm {

// AL = char A

mov AL, byte ptr A

```



```

// AX = AL * char B

imul byte ptr B

// AX += char B

add AL, byte ptr B

adc AH, 0

// short E = AX

mov word ptr E, AX
}

cout << endl << "Output: " << endl;

printf("D = (C-A)/B = %d 0x%04X\n", D, D);

printf("E = (A*B)+B = %d 0x%04X\n", E, E);
}

// 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.