ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»

“ВЫСШАЯ ШКОЛА ЭКОНОМИКИ”

Московский институт электроники и математики им. А.Н.Тихонова

ОТЧЁТ
По домашней работе 2
По курсу “Компьютерный практикум”
Вариант 81

ФИО студента	Группа	Дата	Балл
Волков Егор Ильич	БПМ245	08.02.2025	

Москва 2024

Задание А2:

Написать программу, которая вычисляет значение выражения с помощью стандартных арифметических операций языка С и с помощью ассемблерной вставки.

Исходные данные: *x* = -1h, *y* =-1h, *z* = 4h, v = -1h; *x* = 1h, *y* = 1h, *z* =-29h, = -15h;
x, y – байт, z, v – слова v

1. Использовать только 8-и 16-разрядные регистры и команды базового процессора 8086.
2. Произвести вывод полученных результатов:
   - Из языка Си, например v_C,
   - Из ассемблерной вставки, например v_as,
   - Заданных в тестах значений v.
3. Осуществить все выводы полученных значений в 10- и 16-ричной системах. Если результат отрицательный и получен в виде доп. кода, то по нему восстановить само отрицательное число.
4. Отчёт должен содержать номер варианта, постановку задачи, текст программы с комментариями к каждой команде ассемблера и полученные результаты тестов. В отчете обязательно должно быть представлен выданный преподавателем вариант индивидуального задания. Реализация программы должна осуществляться в MS Visual Studio.

81	$v = \frac{2z-3xy-2}{2x+3} - 4$

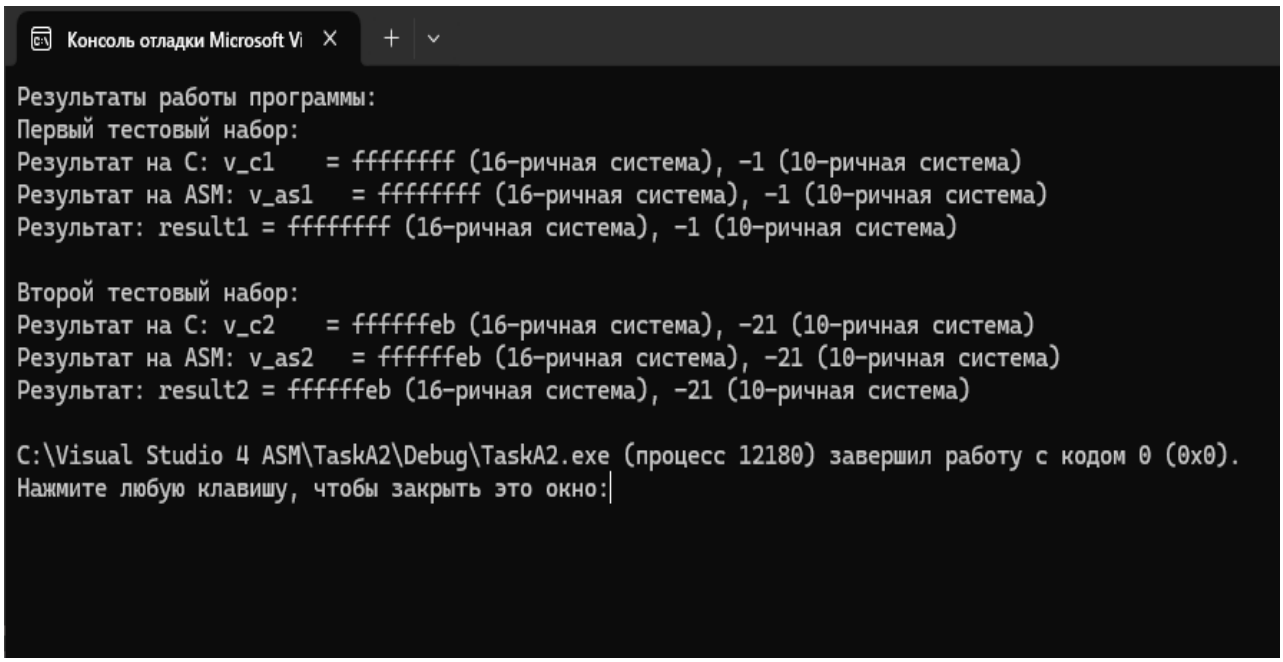
Ассемблерная вставка:

```
__asm {
    // Вычисляем числитель
    mov ax, z; помещаем z в ax
    mov bx, ax; сохранили ax в bx
    mov al, 2; помещаем 2 в al
    cbw; расширяем al в слово
    imul bx; умножаем z на 2
    mov cx, ax; сохраняем(2 * z) в cx
    mov al, x; перемещаем x в al
    mov ah, al; сохраняем al в ah
    mov al, y; перемещаем y в al
    imul ah; умножаем x на y
    cbw; расширяем до слова
    mov dx, ax; сохраняем ax в dx
    mov al, 3; перемещаем 3 в al
    cbw; расширяем до слова
    imul dx; умножаем на три x*y
    sub cx, ax; вычитаем слово 3*x*y из 2z
    sub cx, 2; вычитаем двойку

    // Вычисляем знаменатель
    mov al, x; помещаем x в al
    mov ah, al; сохраняем al в ah
    mov al, 2; перемещаем 2 в al
    imul ah; умножаем 2 на x
    cbw; расширяем до слова
    add ax, 3; прибавляем тройку к 2x

    //Деление числителя на знаменатель
    xchg ax, cx; меняем местами числитель cx и знаменатель ax
    cwd; расширяем до двойного слова
    idiv cx; делим числитель на знаменатель
    sub ax, 4; вычитаем четверку из результата
}
```

Полный код с проверкой значений:

```
#define CRT_NO_WARNINGS
#include <stdio.h>
#include <locale.h>

short int as(char x, char y, short int z) {

    __asm {
        // Вычисляем числитель
        mov ax, z; помещаем z в ax
        mov bx, ax; сохранили ax в bx
        mov al, 2; помещаем 2 в al
        cbw; расширяем al в слово
        imul bx; умножаем z на 2
        mov cx, ax; сохраняем(2 * z) в cx
        mov al, x; перемещаем x в al
        mov ah, al; сохраняем al в ah
        mov al, y; перемещаем y в al
        imul ah; умножаем x на y
        cbw; расширяем до слова
        mov dx, ax; сохраняем ax в dx
        mov al, 3; перемещаем 3 в al
        cbw; расширяем до слова
        imul dx; умножаем на три x*y
        sub cx, ax; вычитаем слово 3*x*y из 2z
        sub cx, 2; вычитаем двойку

        // Вычисляем знаменатель
        mov al, x; помещаем x в al
        mov ah, al; сохраняем al в ah
        mov al, 2; перемещаем 2 в al
        imul ah; умножаем 2 на x
        cbw; расширяем до слова
        add ax, 3; прибавляем тройку к 2x

        //Деление числителя на знаменатель
        xchg ax, cx; меняем местами числитель cx и знаменатель ax
        cwd; расширяем до двойного слова
        idiv cx; делим числитель на знаменатель
        sub ax, 4; вычитаем четверку из результата
    }
}

int main() {

    setlocale(LC_ALL, "rus");

    // Ввод данных для первого теста
    char x1 = -0x1;
    char y1 = -0x1;
    short int z1 = 0x4;
    short int result1 = -0x1;

    // Ввод данных для второго теста
    char x2 = 0x1;
    char y2 = 0x1;
    short int z2 = -0x29;
    short int result2 = -0x15;

    // Первый тестовый набор
    // Считаем результат на C
    short int v_c1 = (((2 * z1) - (3 * x1 * y1) - 2) / (2 * x1 + 3)) - 4;
    // Считаем результат на ASM
    short int v_as1 = as(x1, y1, z1);

    // Второй тестовый набор
    short int v_c2 = (((2 * z2) - (3 * x2 * y2) - 2) / (2 * x2 + 3)) - 4;
    short int v_as2 = as(x2, y2, z2);

    // Выводим результаты
    printf("Результаты работы программы:\n");
    printf("Первый тестовый набор:\n");

    printf("Результат на C: v_c1    = %x (16-ричная система), %d (10-ричная система)\n", v_c1, v_c1);
    printf("Результат на ASM: v_as1   = %x (16-ричная система), %d (10-ричная система)\n", v_as1, v_as1);
    printf("Результат: result1 = %x (16-ричная система), %d (10-ричная система)\n\n", result1, result1);

    printf("Второй тестовый набор:\n");

    printf("Результат на C: v_c2    = %x (16-ричная система), %d (10-ричная система)\n", v_c2, v_c2);
    printf("Результат на ASM: v_as2   = %x (16-ричная система), %d (10-ричная система)\n", v_as2, v_as2);
    printf("Результат: result2 = %x (16-ричная система), %d (10-ричная система)\n", result2, result2);

    return 0;

}
```

Тесты:



Так как во втором тестовом наборе происходило переполнение, пришлось заменить тестовые данные для того, чтобы выполнить задания в рамках требований, не используя 32 битные регистры и еще неизученные команды