

Правительство Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
Высшего профессионального образования  
Национальный исследовательский институт  
**«Высшая школа экономики»**  
Московский институт электроники и математики  
Компьютерная безопасность

**Отчет**  
**по лабораторной работе №2**  
**по курсу «Язык ассемблера»**

**Вариант №33**

Ф.И.О. студента	Номер группы	Дата	Баллы
Николаев Александр Александрович	СКБ191	22.02.2022	

**Задание А2**

**Москва, 2022 г.**

## Задача

Написать программу, которая вычисляет значение выражения с помощью стандартных арифметических операций языка C и с помощью ассемблерной вставки. Проверить на двух тестовых наборах.

Формула:

$$v = \frac{5y - (z + 2) \cdot x}{z - 1} + 1,$$

где  $y, z$  – байты;  $x, v$  – слова

Набор №1:  $x = -1h$ ;  $y = 3h$ ;  $z = 2h$ ;  $v = 14h$ ;

Набор №2:  $x = -2F3h$ ;  $y = 5Bh$ ;  $z = 7Fh$ ;  $v = 309h$

## Текст программы

```
#include <stdio.h>
#include <limits.h>
#include <math.h>
```

```
short int answerAsm(const short int x, const char y, const char z) {
    short int v = 0;
```

```
    __asm {
        mov al, z;           переместить в al z
        cbw;                 расширить z до слова
        add ax, 2;           прибавить к z 2
        imul x;              умножить (z + 2) на x
        mov bx, ax;          переместить младшее слово z в bx
        mov cx, dx;          переместить старшее слово z в cx
        mov al, y;           переместить в al y
        cbw;                 расширить y до слова
        mov dx, 5;           переместить 5 в dx
        imul dx;             умножить y на 5

        sub ax, bx;          вычесть из младшего слова 5y младшее слово (z + 2)x
        sbb dx, cx;          вычесть из старшего слова y старшее слово (z + 2)x

        mov bx, ax;          переместить младшее слово 5y - (z + 2)x в bx

        ;                   вычисление знаменателя
        mov al, z;           переместить в al z
        cbw;                 расширить z до слова
        dec ax;              вычесть из z 1

        xchg ax, bx;         поменять местами ax и bx

        idiv bx;             разделить 5y - (z + 2)x на (z - 1)

        inc ax;              прибавить к слову (5y - (z + 2)x) / (z - 1) 1

        mov v, ax;          переместить в v (5y - (z + 2)x) / (z - 1) + 1
    };
```

```

    return v;
}

short int answerC(const short int x, const char y, const char z) {
    return (5 * y - (z + 2) * x) / (z - 1) + 1;
}

int main() {
    short int x;
    short int v;
    char y;
    char z;
    short int vAsm;
    short int vC;

    int flag = 1;

    printf("\t\tTest results\n");

    x = -0x1; y = 0x3; z = 0x2; v = 0x14;
    printf("\tx = %x, y = %x, z = %x, v = %x \n", x, y, z, v);
    vC = answerC(x, y, z);
    printf("C result:\t\t%d\tor\t%x (base 16)\n", vC, vC);
    vAsm = answerAsm(x, y, z);
    printf("Assembler result:\t%d\tor\t%x (base 16)\n\n", vAsm, vAsm);

    x = -0x2F3; y = 0x5B; z = 0x7F; v = 0x309;
    printf("\tx = %x, y = %x, z = %x, v = %x \n", x, y, z, v);
    vC = answerC(x, y, z);
    printf("C result:\t\t%d\tor\t%x (base 16)\n", vC, vC);
    vAsm = answerAsm(x, y, z);
    printf("Assembler result:\t%d\tor\t%x (base 16)\n\n", vAsm, vAsm);

    printf("Enter 0 to finish: ");
    scanf_s("%d", &flag);

    while(flag) {
        int i;

        while(1) {
            printf("Enter x: ");
            scanf_s("%x", &i);
            if(abs(i) <= SHRT_MAX) {
                x = (short)i;
                break;
            } else {
                printf("Value out of range! Try again\n\n");
            }
        }

        while(1) {
            printf("Enter y: ");
            scanf_s("%x", &i);
            if(abs(i) <= CHAR_MAX) {
                y = (char)i;
                break;
            } else {

```

```

        printf("Value out of range! Try again\n\n");
    }
}

while(1) {
    printf("Enter z: ");
    scanf_s("%x", &i);
    if(abs(i) <= CHAR_MAX && i != 0x1) {
        z = (char)i;
        break;
    } else {
        printf("Value out of range or equals to 1 (division by zero)! Try
again\n\n");
    }
}

while(1) {
    printf("Enter v: ");
    scanf_s("%x", &i);
    if(abs(i) <= SHRT_MAX) {
        v = (short)i;
        break;
    } else {
        printf("Value out of range! Try again\n\n");
    }
}

if(flag) {
    vC = answerC(x, y, z);
    printf("C result:\t\t%d\tor\t%x (base 16)\n", vC, vC);
    vAsm = answerAsm(x, y, z);
    printf("Assembler result:\t%d\tor\t%x (base 16)\n\n", vAsm, vAsm);
}

printf("Enter 0 to finish: ");
scanf_s("%d", &flag);
}

printf("\n Program finished\n");

return 0;
}

```

## Отдельно функция с ассемблерной вставкой (из visual studio)

```
short int answerAsm(const short int x, const char y, const char z) {
    short int v = 0;

    __asm {
        mov al, z;           переместить в al z
        cbw;                 расширить z до слова
        add ax, 2;           прибавить к z 2
        imul x;              умножить (z + 2) на x
        mov bx, ax;          переместить младшее слово z в bx
        mov cx, dx;          переместить старшее слово z в cx
        mov al, y;           переместить в al y
        cbw;                 расширить y до слова
        mov dx, 5;           переместить 5 в dx
        imul dx;             умножить y на 5

        sub ax, bx;          вычесть из младшего слова 5y младшее слово (z + 2)x
        sbb dx, cx;          вычесть из старшего слова y старшее слово (z + 2)x

        mov bx, ax;          переместить младшее слово 5y - (z + 2)x в bx

        ;                   вычисление знаменателя
        mov al, z;           переместить в al z
        cbw;                 расширить z до слова
        dec ax;              вычесть из z 1

        xchg ax, bx;         поменять местами ax и bx

        idiv bx;             разделить 5y - (z + 2)x на (z - 1)

        inc ax;              прибавить к слову (5y - (z + 2)x) / (z - 1) 1

        mov v, ax;          переместить в v (5y - (z + 2)x) / (z - 1) + 1
    };

    return v;
}
```

## Тесты

```
Test results
x = ffffffff, y = 3, z = 2, v = 14
C result:          20      or      14 (base 16)
Assembler result:  20      or      14 (base 16)

x = fffffffd0d, y = 5b, z = 7f, v = 309
C result:          777      or      309 (base 16)
Assembler result:  777      or      309 (base 16)

Enter 0 to finish: 1
Enter x: -1E5
Enter y: 7B
Enter z: 6
Enter v: 384
C result:          900      or      384 (base 16)
Assembler result:  900      or      384 (base 16)

Enter 0 to finish: 0

Program finished
```