



Национальный исследовательский университет «Высшая школа экономики»
Дисциплина «Язык ассемблера»
Специальность 10.05.01 «Компьютерная безопасность»

Отчет
По лабораторной работе №2
по дисциплине «Язык ассемблера»
направления «Компьютерная безопасность»

Вариант 4

Винников Н. В. СКБ-181

Москва, 2021

1. Задание

| Номер варианта | Задание |
|----------------|--|
| 4 | <p>A2.</p> $v = \frac{2x+(z+4)y}{3x-1} - 2,$ $x = 1h, y = -2h, z = -6h, v = 1h$ $x = -6Bh, y = 5F3h, z = 12h, v = -69h$ <p>x – байт, y,z,v - слова</p> |

2. Решение

```
#include <stdio.h>

#define FORMAT_STRING "Test %d\n\nIn decimal: \n\
Ams value: %d\nC value: %d\nExepected value: %d\n\nIn hex: \n\
Ams value: %x\nC value: %x\nExepected value: %x\n\n"

short int c_eva(char x, short int y, short int z) {
    return ((2 * (short)x + (z + 4) * y) / (3 * (short)x - 1) - 2);
}

short ams_eva(char x, short int y, short int z) {
    short vA = 0, dwl = 0, dwr = 0, m = 0;
    __asm {
        ; Вычисление знаменателя
        mov al, x; поместили байт x в AL (AL = 01)
        mov ah, 3;
        imul ah; AL* AH = AX (AX = 0003)
        sub ax, 1; AX - 1 (AX = 0002)
        mov m, ax; перенос слова из AX в m, (3x - 1) в m

        ; Вычисление числителя
        mov ax, z; кладем байт z в регистр AL (AL = 01)
        add ax, 4; z+4
        imul y; (z+4)*y
        mov dwl, ax;
        mov dwr, dx;

        mov al, x; поместили байт x в AL (AL = 01)
        mov ah, 2;
        imul ah; AL* AH = AX (AX = 0002)
        cbw;

        add ax, dwl;
        adc dx, dwr;

        idiv m;
        sub ax, 2;
```

```

        mov vA, ax;

    }
    return vA;
}

int main() {
    short vA = ams_eva(0x1, -0x2, -0x6),
           vC = c_eva(0x1, -0x2, -0x6), v = 0x1;
    printf(FORMAT_STRING, 1, vA, vC, v, vA, vC, v);

    vA = ams_eva(-0x6B, 0x5F3, 0x12),
           vC = c_eva(-0x6B, 0x5F3, 0x12), v = -0x69;
    printf(FORMAT_STRING, 2, vA, vC, v, vA, vC, v);
    return 0;
}

```

```

In decimal:
Ams value: 1
C value: 1
Exepted value: 1

```

```

In hex:
Ams value: 1
C value: 1
Exepted value: 1

```

```

AMS result: v=1
Result in C: v=1
Exepted result: v=1
AMS result: v=ffffff97
Result in C: v=ffffff97
Exepted result: v=ffffff97

```