

Национальный исследовательский университет  
Высшая школа экономики  
Московский институт электроники и математики

Департамент прикладной математики  
кафедра компьютерной безопасности

Отчет по лабораторной работе №2 по дисциплине  
"Языки ассемблера"

**Вариант:** 30

$$V = 3 - \frac{y(x+2)-1}{z+2}$$

**Тест 1-й:**  $x = -3h$ ,  $y = 3h$ ,  $z = -3h$ ,  $V = -1h$

**Тест 2-й:**  $x = 7Eh$ ,  $y = 4000h$ ,  $z = -70h$ ,  $V = 4A7Ch$

Выполнил:  
Фролов О.В.

Проверил:  
Воронцова Т.Д.

Москва, 2023

---

# Оглавление

<b>1</b>	<b>Задание А2</b>	<b>3</b>
1.1	Тесты согласно лабораторной работе . . . . .	6
1.2	Тесты на основе собственных введенных значений . . . . .	6
1.3	Проблема с выводом отрицательных чисел . . . . .	6

# Домашнее задание 1.

---

## Задание А2

1.  $x=-3h$ ,  $y=3h$ ,  $z=-3h$ ,  $V=-1h$

2.  $x=7Eh$ ,  $y=4000h$ ,  $z=-70h$ ,  $V=4A7Ch$

$$V=3 - \frac{y(x+2)-1}{z+2}$$

$x, z$  - байты,  $y, V$  - слова.

```
1 #define _CRT_SECURE_NO_WARNINGS
2 #include <stdio.h>
3 #include <string.h>
4 #include <locale.h>
5 #include <ctype.h>
6
7
8 void assm(short int v_c, short int y, char x, char z) {
9     short int v_as;
10    __asm {
11        // chislitel
12        mov al, x //pomechaem x v AL (AL=13)
13        cbw //expand [byte]AL do [word]AX (AX=0013)
14        inc ax //x+1
15        inc ax //x+2
16        imul y //y*(x+2) (DX:AX)
17        sub ax, 1 //vishislyaem mladshee slovo
18            chislitelya (AX)
19        sbb dx, 0 //vichislyaem starshee slovo
20            chislitelya (DX)
21        //znamenatel
22        mov bx, ax //sochranyaem mladshee slovo
23            chislitelya v BX
24        mov al, z //pomechaem z v AL
25        cbw //expand [byte]AL do [word]AX
26        add ax, 2 //vichislyaem znam v AX
27        xchg ax, bx //pomechaem mladshee slovo
28            chislitelya v AX,
29        //znam - v BX
```

```

26         idiv bx //delim chisl na znam (AX – chastnoe, DX
           – ostatok)
27     mov cx, 3 //pomechaem 3 v CX
28     sub cx, ax //vichli is CX soderzhimoe AX, t.e.
           znachenie drobi
29     mov v_as, cx //resultat pomestili v v_as
30 }
31 printf("Resultat na Assembler'e: \n %d (10–system)
       ili %x (16–system)\n", v_as, v_as);
32 printf("%d", v_as);
33 printf("Resultat na C: \n %d (10–system) ili %x
       (16–system)\n", v_c, v_c);
34 }
35
36 int main() {
37     //connect russian localization
38     setlocale(LC_ALL, "rus");
39     //variables
40     int i, own_input=0;
41     short int y, v_c;
42     char x, z;
43     int start = 1;
44     while (start == 1) {
45         printf("\n\nWould you like to input your own
           values? (0/1): ");
46         if (scanf("%d", &own_input)) {
47             if (own_input == 1) {
48                 printf("\n\nInput values of X, Y, Z: ");
49                 scanf("%hhx %hx %hhx", &x, &y, &z);
50                 while (z==–0x2) {
51                     printf("\n\nZ can't be equal –0x2!");
52                     printf("\nInput values of X, Y, Z: ");
53                     scanf("%hhx %hx %hhx", &x, &y, &z);
54                 }
55             }
56             if (own_input == 0) {
57                 //communication with user
58                 printf("Input number of test: \n");
59                 scanf("%d", &i);
60                 switch (i) {
61                     case 1:
62                         x = –0x3;
63                         y = 0x3;

```

```

64             z = -0x3;
65             break;
66         case 2:
67             x = 0x7E;
68             y = 0x4000;
69             z = -0x70;
70             break;
71     }
72 }
73 v_c = 3 - ((y * (x + 2) - 1) / (z + 2));
74 asmm(v_c, y, x, z);
75 //cycle of checking
76 printf("\n\nAgain? (0/1): ");
77 scanf("%d", &start);
78 }
79 else {
80     printf("\n\nIncorrect input!");
81 }
82 }
83 return 0;
84 }

```

### Frolov\_A2.c

Принцип работы: Сначала пользователь выбирает режим: можно использовать уже заранее заготовленные данные (в рамках условий лабораторной работы), а можно ввести собственные значения с ограничением на z, т.к. она не должна быть равна -2.

Если пользователь выбрал использование заготовленных тестовых данных, то он выбирает набор (всего их 2). Далее он получает вывод на экране результатов работы программы.

Если пользователь выбрал ввод собственных значений, то он должен ввести x, y, z в формате 0xffff через пробел в одну строку. Далее происходят вычисления, и они же выводятся на экран.

После всех процедур пользователю поступает вопрос о том, нужно ли повторить работу программы с начала, или же можно завершить выполнение программы.

## 1. Тесты согласно лабораторной работе

```
Would you like to input your own values? (0/1): 0
Input number of test:
1
Resultat na Assembler'e:
-1 (10-system) ili ffffffff (16-system)
-1Resultat na C:
-1 (10-system) ili ffffffff (16-system)

Again? (0/1): 1

Would you like to input your own values? (0/1): 0
Input number of test:
2
Resultat na Assembler'e:
19068 (10-system) ili 4a7c (16-system)
19068Resultat na C:
19068 (10-system) ili 4a7c (16-system)

Again? (0/1):
```

## 2. Тесты на основе собственных введенных значений

```
Would you like to input your own values? (0/1): 1

Input values of X, Y, Z: -0x5 0x8E -0x5
Resultat na Assembler'e:
-139 (10-system) ili fffffff75 (16-system)
-139Resultat na C:
-139 (10-system) ili fffffff75 (16-system)

Again? (0/1):
```

## 3. Проблема с выводом отрицательных чисел

Дело в том, что ассемблер представляет отрицательные числа в дополнительном коде, поэтому по итогу мы видим, например, ffff вместо -1.