



Национальный исследовательский университет «Высшая школа экономики»

Дисциплина «»

Специальность 10.05.01 «Компьютерная безопасность»

Отчет
По лабораторной работе №4
по дисциплине «Языки Ассемблера»
направления «Компьютерная безопасность»

Вариант 4

Винников Н. В. СКБ-181

Москва, 2021

4. A1. a) 3335 б) C8h 2Dh

Винников Н.

A2. $v = \frac{2x + (z + 4)y}{3x - 1} - 2$; $x = 1h, y = -2h, z = -6h, v = 1h$; $x = -6Bh, y = 5F3h, z = 12h, v = -69h$; x – байт, y, z, v – слова.

A3. Дана строка из трех десятичных цифр. Если вторая и третья цифры — частное и остаток от деления первой цифры на 5, то уменьшить третью цифру на 1 ('0' не изменять), иначе — все цифры заменить на первую.

A4. Даны массивы A и B из 16 байтов. Если элемент массива A больше соответствующего элемента массива B (числа знаковые), то обменять эти элементы местами. Сосчитать количество обменов. В массив C поместить адреса (смещения) этих элементов.

Даны массивы A и B из 16 байтов. Если элемент массива A больше соответствующего элемента массива B (числа знаковые), то обменять эти элементы местами. Сосчитать количество обменов. В массив C поместить адреса (смещения) этих элементов.

Выполнить задание A4 в двух вариантах: A4.1 и A4.2. В первом варианте все описания заданных массивов полностью соответствуют постановке задания. Во втором варианте использовать расширенные описания исходных массивов: байтовые. байтовые массивы становятся массивами из слов, массивы слов становятся массивами из двойных слов.

```
#include <stdio.h>
#include <string.h>
#include <locale.h>
#include <ctype.h>
#include <cstdlib>
#include <iostream>
```

```
int main()
{
    setlocale(LC_ALL, "ru");
    setlocale(LC_NUMERIC, "en");
    unsigned int res;
    char a[16] = { -1,-2,-3,-4,-6,-7,0,1,2,3,4,5,6,7,8,9 };
    char b[16] = { -3,-2,-1, 0,-1,-2,3,1,1,2,2,6,6,8,8,9 };
    unsigned int c[16];

    printf("Исходные массивы:\n");
    for (unsigned int i = 0; i < 16; i++)
    {
        printf("A.%.2d=%d\tB.%.2d=%d\n", i, (int)a[i], i, (int)b[i]);
    }

    _asm {
        lea esi, a           // esi - указатель на элементы массива a
        lea edi, b           // edi - указатель на элементы массива b
        xor ebx, ebx         // ebx - счетчик элементов в массиве c
        xor ecx, ecx         // ecx = 0, номер элемента массива число
элементов массивов
        m2 :
            mov al, [esi]     // Берем очередной элемент массива a
            cmp al, [edi]     // Сравниваем с очередным элементом
массива b
            jle m1            // Меньше или равно, ничего делать
не надо
            // a > b
            mov c[ebx*4],esi // Если надо сохранять абсолютный адрес массива a
```

```

//          mov    byte ptr c[ebx * 4], cl          // Заносим смещение
элементов относительно начала массива в массив c
        inc     ebx                                // ebx++ - следующий элемент
массива c
        mov     ah, [edi]                          // меняем местами элементы массивов a и b
        mov[edi], al
        mov[esi], ah
m1 :
        inc     esi                                // esi++ следующий элемент массива
a
        inc     edi                                // edi++ следующий элемент массива
b
        inc     ecx                                // ecx++ увеличение счетчика
        cmp     cl, 16                             // cl = 16?
        jnz     m2                                 // Нет. Следующая итерация
        mov     res, ebx                           // res = число обменов
    }

printf("Результат:\nЧисло обменов: %d\n", res);
for (unsigned int i = 0; i < 16; i++)
{
    printf("A.%.2d=%d\tB.%.2d=%d\n", i, (int)a[i], i, (int)b[i]);
}

for (unsigned int i = 0; i < res; i++)
{
    printf("C.%.2d=%X\n", i, c[i]);
}
system("pause");
}

```

Исходные массивы:

```

A.00=-1 B.00=-3
A.01=-2 B.01=-2
A.02=-3 B.02=-1
A.03=-4 B.03=0
A.04=-6 B.04=-1
A.05=-7 B.05=-2
A.06=0 B.06=3
A.07=1 B.07=1
A.08=2 B.08=1
A.09=3 B.09=2
A.10=4 B.10=2
A.11=5 B.11=6
A.12=6 B.12=6
A.13=7 B.13=8
A.14=8 B.14=8
A.15=9 B.15=9

```

Результат:

Число обменов: 4

```

A.00=-3 B.00=-1
A.01=-2 B.01=-2
A.02=-3 B.02=-1
A.03=-4 B.03=0
A.04=-6 B.04=-1
A.05=-7 B.05=-2
A.06=0 B.06=3
A.07=1 B.07=1
A.08=1 B.08=2
A.09=2 B.09=3
A.10=2 B.10=4
A.11=5 B.11=6
A.12=6 B.12=6
A.13=7 B.13=8
A.14=8 B.14=8
A.15=9 B.15=9

```

C.00=BEF7C0

C.01=BEF7C8

C.02=BEF7C9

C.03=BEF7CA

Для продолжения нажмите любую клавишу . . .