

Национальный исследовательский университет
Высшая школа экономики
Московский институт электроники и математики

Департамент прикладной математики
кафедра компьютерной безопасности

Отчет по лабораторной работе №4.2 по дисциплине
"Языки ассемблера"

Вариант: 30

Выполнил:
Фролов О.В.

Москва, 2023

Оглавление

1	Задание А4.2	3
1.1	Тесты	5

Домашнее задание 1.

Задание А4.2

Дан массив А из 16 слов. Если три последовательных элемента составляют невозрастающую последовательность, то скопировать средний элемент в массив слов В, а адрес (смещение) этого элемента — в массив С (как и в предыдущей работе массив двойных слов). Сосчитать количество таких элементов. (В последовательности 3,2,0,1,7,7,2,1,0 таких элементов четыре.)

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <string.h>
4  #include <locale.h>
5  #include <ctype.h>
6
7  int main() {
8      setlocale(LC_ALL, "rus");
9
10     int start = 1, flag=0;
11     while (start == 1) {
12         short int a[16], b[14] = { 0,0,0,0,0,0,0,0,0,0,0,0,0,0 }; //14, а не 16, т.к. средних элементов может быть максимум 14 при 16 исходных
13         int c[16] = { 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 }; //под адреса
14         int res = 0;
15
16         input:
17         printf("\n\nInput numbers (16 numbers): ");
18         scanf("%d %d %d %d %d %d %d %d %d %d %d %d %d %d %d", &a[0], &a[1], &a[2], &a[3], &a[4], &a[5], &a[6], &a[7], &a[8], &a[9], &a[10], &a[11], &a[12], &a[13], &a[14], &a[15]);
19         getchar();
20
21         asm {
22             lea esi, a //адрес начала массива a поместили в esi
23             lea edi, b //адрес начала массива b поместили в edi
24             lea edx, c //адрес начала массива c поместили в edx
25             mov cx, 14 //макс. число повторений в цикле поместили в cx, 14 т.к. последние два элемента не образуют новых последовательностей из трех
26
27             first:
28             mov al, [esi] //элемент a[i] поместили в регистр al (как байт, поэтому +1 дальше идет для перехода к след. элементу)
29             mov bl, [esi + 2] //элемент a[i+1] поместили в регистр bl(как байт)
30
31             cmp al, bl //сравнили a[i] и a[i+1]
32             jl second //переходим на second, если первый элемент < второго
33
34             mov al, [esi+4] //элемент a[i+2] поместили в регистр al
35             cmp bl, al //сравнили a[i+1] и a[i+2]
36             jl second //переходим на second, если второй элемент < третьего
37
38             //скопировать средний элемент (т.е. bl) в массив В, а адрес (смещение) этого элемента в С
39             mov [edi], bl //поместили средний элемент a[i+1] в b[i]
40             add esi, 4 //изменили адрес на адрес следующего элемента
41             mov [edx], esi //поместили адрес (смещение) a[i+1] в c[i]
42             sub esi, 4 //вернули прежний адрес
43             add edi, 2 //увеличили i для b[i]
```

```

43      add edi, 2 //увеличили i для b[i]
44      add edx, 4 //увеличили i на 1 для c[i]
45      add res, 1 //увеличили кол-во таких элементов
46      second:
47      add esi, 2 //увеличили i для a[i]
48      dec cx //cx=cx-1
49      cmp cx, 0
50      jne first //уйти на first, если cx не равно 0
51      nop
52  }
53  printf("Input massive: ");
54  for (int i = 0; i < 16; ++i) {
55      printf(" %d ", a[i]);
56  }
57  printf("\nResult massive: ");
58  for (int i = 0; i < 16; ++i) {
59      printf(" %x ", c[i]);
60  }
61  printf("\nResult is %d", res);
62  printf("\n\nAgain(0/1): ");
63  if (scanf("%d", &start)) {
64      getchar();
65      if (start == 1)
66          start = 1;
67      else
68          start = 0;
69  }
70  }
71  exit(0);
72  }

```

Принцип работы: Сначала пользователь вводит 16 чисел.

Далее выполняется ассемблерная вставка: в регистры esi, edi, edx помещаются адреса, с которых начинаются массивы a, b, c соответственно. В cx помещается максимальное кол-во итераций последующих процедур (14, т.к. чисел всего 16, а проверяются тройками). Далее в al помещается первое число в текущей итерации (в зависимости от адреса, лежащего в esi), а в bl второе аналогично, они сравниваются, если первое оказывается меньше второго, то происходит переход на second, где увеличивается счетчик итераций, лежащий в cx, и если он не равен 0, то происходит переход на first, но перед этим увеличивается адрес esi для перехода к следующей тройке элементов. Если первое число оказалось больше или равно второму, то в al помещается третье число, а дальше сравнивается bl и al, если второе число окажется меньше третьего, то произойдет переход на second, иначе средний элемент будет помещен в массив b, а адрес (смещение) этого элемента в массив c, счетчик этих элементов res будет увеличен на 1, а дальше переход на second.

После всех процедур пользователю вернется количество искомых элементов, а далее поступает вопрос о том, нужно ли повторить работу программы, или же можно завершить выполнение программы.

1. Тесты

```
D:\Games\Frolov-A4.2\Debug\Frolov-A4.2.exe

Input numbers (16 numbers): 0 1 2 3 4 5 6 7 8 15 14 13 12 11 10 9
Input massive: 0 1 2 3 4 5 6 7 8 15 14 13 12 11 10 9
Result massive: 10ffdfa 10ffdfc 10ffdfc 10ffe00 10ffe02 0 0 0 0 0 0 0 0 0 0
Result is 5

Again(0/1)?: 1

Input numbers (16 numbers): 0 1 2 3 4 5 6 7 8 9 10 11 12 15 14 13
Input massive: 0 1 2 3 4 5 6 7 8 9 10 11 12 15 14 13
Result massive: 10ffe02 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Result is 1

Again(0/1)?: 1

Input numbers (16 numbers): 15 8 7 11 10 9 6 13 14 12 0 1 2 3 4 5
Input massive: 15 8 7 11 10 9 6 13 14 12 0 1 2 3 4 5
Result massive: 10ffde8 10ffdee 10ffdf0 10ffdf8 0 0 0 0 0 0 0 0 0 0 0
Result is 4

Again(0/1)?: 1
```