Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное   
образовательное учреждение высшего образования   
«Самарский национальный исследовательский университет   
имени академика С.П. Королева»

Институт информатики, математики и электроники

Факультет информатики

Кафедра технической кибернетики

Отчет по лабораторной работе №4

Дисциплина: «Системное программирование и архитектура ЭВМ»

Выполнил: Абрамов Д.О.

Группа: 6208-010302D

Самара, 2021

**Задание на лабораторную работу**

1) В программе необходимо реализовать функцию обработки элементов массива используя команды сравнения, переходов и циклов на встроенном ассемблере.

2) Элементы массива – целые числа.

3) Результат – целочисленный, возвращается из функции регистре eax.

4) Указатель на первый элемент массива передаётся в качестве параметра функции.

5) В программе реализовать вывод массива и результата обработки на экран.

6) В качестве комментария к каждой строке необходимо указать, какое действие выполняет команда относительно массива.

**Вариант 1** Найти минимальный элемент в массиве A={a[i]} и подсчитать количество его вхождений.

**Код программы**

#include <iostream>

#include <algorithm>

//Найти минимальный элемент в массиве A={a[i]} и подсчитать количество его вхождений

using namespace std;

int main()

{

int res = 1;

int arr[5] = { 2, -5, -1, -5, 1};

int n = 5;

int A = \*arr;

for (int i = 1; i < n; ++i) {

if (A == arr[i]) ++res;

if (arr[i] < A) {

A = arr[i];

res = 1;

}

}

cout << "res c++: " << A << " " << res << endl;

int error;

int res\_asm;

int res\_asm2 = 1;

\_\_asm {

xor esi, esi

xor edi, edi

mov ecx, n

mov ebx, arr[esi \* 4]

mov res\_asm, ebx

inc esi

dec ecx

mov edx, res\_asm2

begin\_loop :

mov eax, arr[esi \* 4]

cmp eax, ebx

je ravno

jmp neravno;

ravno:

add edx, 1

mov res\_asm2, edx

neravno:

cmp eax, ebx

jge loop\_end

mov ebx, eax

mov res\_asm, eax

mov edx, 1

mov res\_asm2, 1

loop\_end:

inc esi // переходим к следующему элементу

loop begin\_loop // повторяем цикл для всех элементов массива

}

cout << "res asm: " << res\_asm << " " << res\_asm2 << endl;

system("pause");

return 0;

}

