Міністерство освіти і науки України

Одеський національний політехнічний університет

Iнститут компьютерних систем

Кафедра компьютерних інтеллектуальних систем і мереж

Лабораторна робота №5

З дисципліни: “Архiтектура комп’ютерiв”

По темi: “Організація циклів і робота з одномірними масивами”

Зробив

Ст. Борщов М.I.

Групи АМ-182

Перевiрили:

Шапорiн Р.О.

Одеса 2020

**Мета:** Ознайомитися з організацією циклів і навчитися працювати з одномірними масивами.

**Завдання:**

Завдання до лабораторної роботи.

1. Вибрати свій варіант обробки масиву.



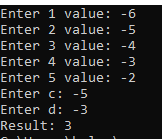
2. Написати модуль обробки елементів мовою Асемблера.

3. Написати на базовій алгоритмічній мові програму коректного вводу вихідних даних (з контролем

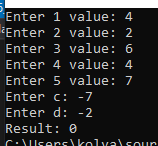
припустимого діапазону) і виводу отриманого результату.

4. Вмонтувати виклик модуля мовою Асемблера в програму на базовій алгоритмічній мові.

5. Зробити тестові перевірки, зробити аналіз результатів.



Перевірка С <= Елемент масиву <= D



Якщо негативних елементів нема

6. Занесіть до протоколу:

- вихідні дані й отримані результати обчислень у шістнадцятирічній формі, а також їхнє місце розташування

в оперативній пам'яті і їхні еквіваленти в десятковій формі (для можливості перевірки обчислень);

- текст програми і її опис.

Код програми

#include <iostream>

using namespace std;

short result = 0;

short func(short arr, short c, short d)

{

\_asm {

mov dx, c

mov cx, arr // загружаю элемент массива в cx

cmp cx, 0 // сравниваю сх с 0

JG miss // если сх > 0, ничего не делаю, беру следующий элемент массива

mov bx, c // загружаю в bх с

cmp cx, bx // сравниваю cx c bx (с)

JGE nextStep // если элемент >= c, переходим дальше JGE = jump greater equals cx >= bx

JMP miss // иначе беру следующий элемент массива

nextStep: // переход "дальше"

mov bx, d // загружаю в bx d

cmp cx, bx // сравниваю cx с bx (d)

JLE lastStep // если элемент <= d, переход на lastStep JLE = jump less equal cx <= bx

jmp miss

lastStep: // lastStep

add result,1 // добавляю к счётчику 1

JMP miss

JMP miss // прыгаю на следующий элемент массива

miss: // следующий элемент массива

}

return result;

}

int main()

{

short enter = 0, c = 0, d = 0;

short mas[5];

for (int i = 0; i < 5; ++i) {

cout << "Enter " << i + 1 << " value: ";

cin >> enter;

mas[i] = enter;

}

cout << "Enter c: ";

cin >> c;

cout << "Enter d: ";

cin >> d;

for (int i = 0; i < 5; i++) {

func(mas[i], c, d);

}

cout << "Result: " << result;

return 0;

}

**Висновок:** У цій лабораторній роботі я навчився організовувати цикли, та працювати з одномірними масивами на ЯП ASSEMBLER, Перевіряти чи виконуються умови перевірки, зрівнювати різні змінні.