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. Алькавелани М.

Групи АМ-181

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 С <= Елемент масиву <= D нема, то буде виведено лише нулi

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 respos = 0, resneg = 0, reszero = 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

JE zeroN // если сх = 0,

JL lowN // если сх < 0,

JG highN // если сх > 0,

lowN:

mov bx, c // загружаю в bх с

cmp cx, bx // сравниваю ax c bx (с)

JGE lowStep

JMP miss // иначе беру следующий элемент массива

highN:

mov bx, c // загружаю в bх с

cmp cx, bx // сравниваю cx c bx (с)

JGE highStep

JMP miss // иначе беру следующий элемент массива

lowStep: // переход "дальше"

mov bx, d // загружаю в bx d

cmp cx, bx // сравниваю aх с bx (d)

JLE lastlow

jmp miss

highStep: // переход "дальше"

mov bx, d // загружаю в bx d

cmp cx, bx // сравниваю cх с bx (d)

JLE lasthigh

jmp miss

lastlow: // lastStep

add resneg,1 // добавляю к счётчику 1

JMP miss

lasthigh: // lastStep

add respos, 1 // добавляю к счётчику 1

JMP miss

zeroN:

add reszero, 1 // добавляю к счётчику 1

JMP miss

JMP miss // прыгаю на следующий элемент массива

miss: // следующий элемент массива

}

return respos, resneg, reszero;

}

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 << "Positive numbers: " << respos << ". Negative numbers: " << resneg << ". Zero numbers: " << reszero << '.';

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.