**“ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА”**

**(СПбГУТ)**

**Факультет инфокоммуникационных Сетей и систем (иксс)**

**кафедра программной инженерии и вычислительной техники (пи и вт)**

Дисциплина: «Машинно-зависимые языки программирования»

Лабораторная работа №4.

**Тема: «Обработка одномерных массивов»**

**вариант №1**

Выполнил: Студент группы ИКПИ-95

Алюшин В.В.

Подпись\_\_\_\_\_\_\_\_\_\_\_\_

Принял: Коробов С.А.

Подпись\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_\_»\_\_\_\_\_\_\_\_ 2020

2020г

**ЦЕЛЬ РАБОТЫ:**

Задав одномерный массив целочисленных данных **А** в одном из заданных форматов (unsigned char — **BYTE**, unsigned short int — **WORD**, char — **SHORTEST**, short int — **INTEGER** или long int — **LONGINT**), реализовать обработку массива, как указано в варианте. Длина массива **N**. Исходные данные задать самостоятельно, учитывая формат элементов массива **А**.  
  
В программе на **C** должны быть предусмотрены функции ввода - вывода элементов массива **А** и его обработки. Исходные данные должны вводиться корректно и быть приближенными к максимальновозможным для данного типа данных. Тип результата определяется из контекста задачи.

**Найти сколько элементов массива A={a[i]} удовлетворяет условию с <= a[i] <= d**

**Таблица Результатов**

|  |  |  |
| --- | --- | --- |
| **c** | **d** | **результат** |
| **-100** | **100** | **10** |
| **3** | **6** | **1** |
| **1** | **22** | **2** |
| **-3** | **3** | **6** |

**Код:**

**Файл main.c:**

#include <conio.h>

#include <stdio.h>

#include <stdint.h>

#include <stdlib.h>

#define arr\_size 256

int64\_t c, d;

extern "C"

{

extern \_\_int64 my\_array[arr\_size];

uint64\_t asmF(int64\_t c, int64\_t d);

}

void cMode()

{

uint64\_t counter = 0;

for(uint64\_t i = 0; i < arr\_size; i++)

{

if (c <= my\_array[i] && my\_array[i] <= d)

{

counter++;

}

}

printf("Result: %I64u\n", counter);

}

void asmMode()

{

uint64\_t sum = asmF(c, d);

printf("Result: %I64u\n", sum);

}

int main()

{

for (int64\_t i = 0; i < arr\_size; i++)

{

my\_array[i] = (int64\_t)rand() % 21 - 10;

printf("%I64d\n", my\_array[i]);

}

while (1)

{

printf("Enter c: ");

scanf("%I64d", &c);

printf("Enter d: ");

scanf("%I64d", &d);

cMode();

asmMode();

}

\_getch();

return 0;

}

**Файл ASM (ASM):**

.data

public my\_array

my\_array dq 256 DUP(0)

arr\_size dq 256

.code

asmF proc

mov r13, rdx

xor r10, r10 ;counter = 0

xor r8, r8 ;i = 0

;r12 = arr\_size

mov r12, arr\_size

@L1whileIf:

cmp r8, r12

je @L2End

;if

mov rax, 8

xor rdx, rdx

mul r8

lea rbx, my\_array

add rax, rbx

mov rbx, [rax]

cmp rcx, rbx ;c <= a[i]

jg @L3EndIf

cmp rbx, r13 ;a[i] <= d

jg @L3EndIf

inc r10 ;l++

@L3EndIf:

inc r8 ; --i

jmp @L1whileIf

@L2End:

mov rax, r10

ret

asmF endp

end