**Лабораторная работа №4**

**ОБРАБОТКА ОДНОМЕРНЫХ МАССИВОВ**

**Студент гр. ИКПИ-22**

**Нестеренко Н.С.**

**Постановка задачи**

**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 максимально возможным для данного типа данных. Тип результата определяется из контекста задачи.

**ПОРЯДОК РАБОТЫ:**

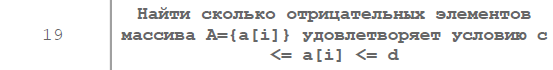
• внимательно изучить свой вариант обработки элементов массива;

• написать на языке Cи программу ввода исходных данных (с контролем допустимого диапазона), обработки элементов массива и вывода полученного результата;

• написать модуль обработки элементов массива на языке Ассемблера;

• встроить вызов этого модуля в программу на языке Си;

**ВАРИАНТ ЗАДАНИЯ:**



**Разработка алгоритма**

1. Входные данные:

signed short (word): a, c, d

unsigned char (byte): n, count

**Таблица идентификаторов**

|  |  |  |  |
| --- | --- | --- | --- |
| **N** | **Обозначение в задаче** | **Размер** | **Назначение** |
| **1** | **a** | signed short (word)  16 бит | **Входные данные** |
| **2** | **d** |
| **3** | **c** |
| **4** | **n** | unsigned char (byte)  8 бит |
| **5** | **count** | unsigned char (byte)  8 бит | **Выходные данные** |

**Контрольный расчет**

Результаты вычислений приведены ниже в таблице вычислений.

|  |  |  |  |
| --- | --- | --- | --- |
| **Назначение набора данных** | **Набор данных**  **(a[3], c, d)** | **Результаты вычислений**  **(count)** | |
| **аналитически** | **asm** |
| **Контрольный набор** | 8307 -29387 -24685  -25000 0 | 1 | 1 |
| -21198 6549 17324  -21198 -21198 | 1 | 1 |
| -4759 -8953 -29947  0 -3000 | 0 | 0 |

**Программа на языке asm (nasm)**

section .data

    extern a, c, d, n, count

section .text

global asm\_func

asm\_func:

    mov ebx, 0

    @for\_loop:

    mov ax, [a + ebx\*2]

    inc bx

    cmp ax, 0

    jl @neg\_true

    jmp @else

    @neg\_true:

    mov cx, [c]

    cmp cx, ax

    jg @else

    mov cx, [d]

    cmp ax, cx

    jg @else

    movzx cx, *byte*[count]

    inc cx

    mov [count], cx

    @else:

    cmp bl, [n]

    jne @for\_loop

    ret

**Программа на языке Cи**

#include <stdlib.h>

#include <stdio.h>

#include <stdint.h>

#include <time.h>

#define N 10

*int16\_t* a[N], c, d;

*uint8\_t* n = N, count = 0;

extern *void* asm\_func();

// вариант 19

// кол-во отрицательных элементов

// с <= a[i] <= d

*int* main(*void*) {

    srand(time(NULL));

    printf("A = ");

    for(*int* i = 0; i < n; i++) {

        a[i] = -32768 + rand() % 65536;

        printf("%hd ", a[i]);

    }

    printf("\ninput c, d (-32768 to 32767): ");

    scanf("%hd %hd", &c, &d);

    asm\_func();

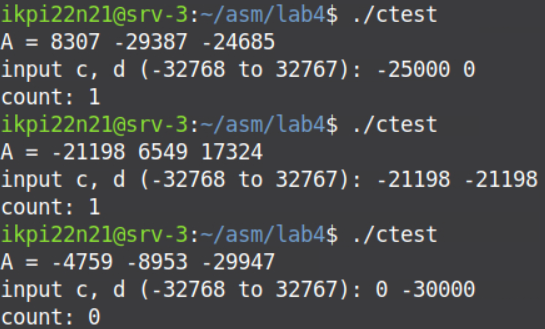
    printf("count: %hhd\n", count);

    return 0;

};

**Результаты работы программы**

Далее предоставлены результаты выполнения программы с выбранным для тестирования контрольным набором исходных данных.



**Выводы**

Тот факт, что результаты расчетов на ассемблере, совпадают с результатами аналитических расчётов, свидетельствуют о том, что программа составлена правильно.