Федеральное государственное бюджетное образовательное учреждение высшего образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича»

Кафедра «Программная инженерия и вычислительная техника»

«Машино-зависимые языки программирования»

Отчет

по лабораторной работе №4

«Обработка одномерных массивов»

|  |  |
| --- | --- |
|  | Выполнил |
|  | студент группы ИКПИ-14 |
|  | А.В.Крылов |
|  |  |
|  | Проверил |
|  | Ст. преподаватель |
|  | А.О.Овчинников |
|  |  |

2022 г.

1. Задание

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.

Вариант №11

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

2. Текст программы

2.1. Модуль main.c

#include <stdio.h>

#define N 5

signed char A[N];

signed char d, c;

extern signed char afunc();

int main() {

puts(

"Zadacha №11, Krylov Artem Vyacheslavovich, IKPI-14\n"

"Nayti skolko elem massiva A udovletroryayut usloviyu:\n"

"d <= A[i] <= c, [-128; 127]\n");

puts("Vvedite d i c:");

scanf("%hhi%hhi", &d, &c);

puts("Vvedite 5 elem A:");

for (int i = 0; i < N; ++i) scanf("%hhi", &A[i]);

printf("ASM: %hhi\n", afunc());

signed char count = 0;

for (int i = 0; i < N; ++i)

if (d <= A[i] && A[i] <= c) ++count;

printf("C: %hhi\n", count);

return 0;

}

2.2. Модуль afunc.asm

extern A, d, c

global afunc

section .text

afunc:

push rbp

mov rbp, rsp

xor dl, dl

mov al, [d]

mov bl, [c]

mov ecx, 4

funcloop:

cmp al, [A + ecx]

jg loopend

cmp bl, [A + ecx]

jl loopend

inc dl

loopend:

dec cl

cmp cl, 0

jge funcloop

mov al, dl

mov rsp, rbp

pop rbp

ret

2.3. Модуль Makefile

# common

C = gcc -Wall -Wextra -Wpedantic -fsanitize=address,undefined

A = yasm -f elf64

# compile-only args

CO = -c

S = src/

O = obj/

OBJS = $(O)main.o $(O)afunc.o

all: $(OBJS)

$(C) $(OBJS) -fno-pie -no-pie -o L4

$(O)main.o: $(S)main.c

$(C) $(CO) $(S)main.c -o $(O)main.o

$(O)afunc.o: $(S)afunc.asm

$(A) $(S)afunc.asm -o $(O)afunc.o

clean:

rm $(O)\*.o && rm L4

3. Сборка проекта

make

4. Выполнение программы

4.1. Запуск программы

./L4

4.2. Входные данные

-127 126

-128 -127 127 126 0

4.3. Ожидаемый результат выполнения

ASM: 3

C: 3

4.4. Результат выполнения

ASM: 3

C: 3

5. Вывод

Результат выполнения программы соответствует ожидаемому результату. Работа выполнена в полном объеме.