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»

|  |  |
| --- | --- |
|  | Выполнила |
|  | студентка группы ИКПИ-25 |
|  | А. Ю. Яскунова |
|  |  |
|  | Проверил |
|  | Ст. преподаватель |
|  | И.Л. Савельев |
|  |  |

2023 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 №26**

Найти произведение первых L отрицательных элементов массива A={a[i]} удовлетворяет условию с <= a[i] <= d

**2. Текст программы**

**2.1 Модуль lab4.c**

#include <stdio.h>

// Функция для ввода элементов массива А

void inputArray(int A[], int N) {

printf("Введите элементы массива:\n");

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

printf("A[%d]: ", i);

scanf("%d", &A[i]);

}

}

// Функция для вывода элементов массива А

void outputArray(int A[], int N) {

printf("Элементы массива:\n");

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

printf("%d ", A[i]);

}

printf("\n");

}

// Функция для вызова модуля обработки на языке Ассемблера

extern void asmArrayFunc();

int N, L, c, d, result\_asm;

int arr[];

int main() {

printf("Введите длину массива N: ");

scanf("%d", &N);

inputArray(arr, N);

printf("Введите количество первых отрицательных элементов L: ");

scanf("%d", &L);

printf("Введите значение c: ");

scanf("%d", &c);

printf("Введите значение d: ");

scanf("%d", &d);

outputArray(arr, N);

int result\_c = 1;

int count = 0;

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

if (arr[i] >= c && arr[i] <= d) {

if (arr[i] < 0) {

result\_c \*= arr[i];

count++;

}

if (count == L) {

break;

}

}

}

printf("Результат на СИ: произведение первых %d отрицательных элементов в промежутке [%d, %d] равно: %d\n", L, c, d, result\_c);

asmArrayFunc();

printf("Результат на ассемблер: произведение первых %d отрицательных элементов в промежутке [%d, %d] равно: %d \n", L, c, d, result\_asm);

return 0;

}

**2.2 Модуль asm\_lab4.asm**

section .data

extern N

extern L

extern c

extern d

extern result\_asm

extern arr

section .text

global asmArrayFunc

asmArrayFunc:

mov ecx, 0 ; счетчик цикла

mov ebx, 1 ; инициализируем произведение единицей

mov edx, 0 ;

cycle:

mov eax, dword [arr + ecx \* 4]

cmp eax, 0

jl check\_conditions

jmp next\_iteration

next\_iteration:

inc ecx

cmp ecx, [N] ; условие выхода из цикла

jng cycle

jmp exit

check\_conditions:

cmp edx, [L]

je exit

cmp eax, [d]

jg next\_iteration

cmp eax, [c]

jl next\_iteration

imul ebx, eax

inc edx

jmp next\_iteration

exit:

mov dword [result\_asm], ebx

ret

**2.3. Модуль makefile**

all:

gcc -c -g -ggdb -o lab4.o lab4.c

nasm -f elf64 -F dwarf -g asm\_lab4.asm

gcc -m64 -o lab4 lab4.o asm\_lab4.o -g -ggdb -fno-pie -no-pie

**3. Сборка проекта**

make

**4. Выполнение программы**

**4.1. Запуск программы**

./lab4

**4.2. Входные данные**

N = 5

L= 3

c = -2500

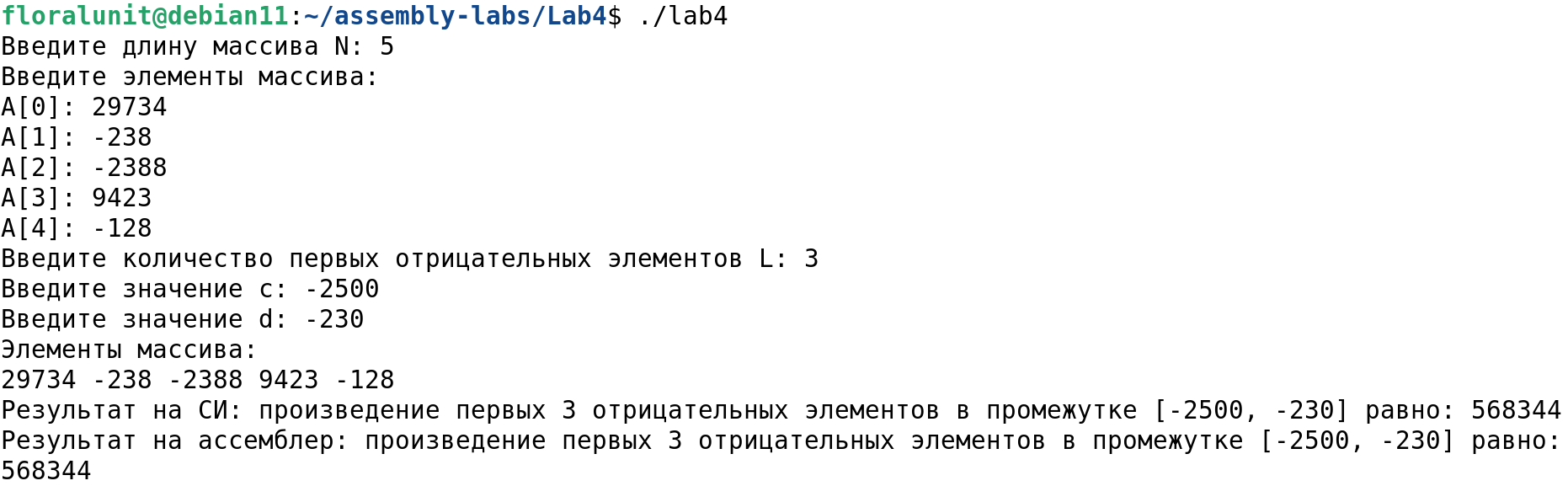
d = -230

{29734, -238, -2388, 9423, -128}

**4.3. Ожидаемый результат выполнения**

568344

**4.4. Результат выполнения**



**5. Вывод**

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