Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет  
информатики и радиоэлектроники»

Инженерно-экономический факультет

Кафедра экономической информатики

ОТЧЁТ

к лабораторной работе №4

на тему

«Одномерные массивы»

по курсу

«Основы алгоритмизации и программирования»

Вариант 9

|  |  |
| --- | --- |
| Выполнил: | А. С. Жевлаков  студент гр. 578104 |
|  |  |
| Проверил: | Е. И. Полоско  ст.преподаватель кафедры ЭИ |

Минск 2025

Цель работы: познакомиться с созданием и обработкой одномерных массивов данных; научиться использовать одномерные массивы в алгоритмах вычисления выражений, сортировки, поиска.

**КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

**Массив** – это совокупность однотипных элементов, имеющих общее имя и расположенных в памяти рядом.

Массив располагается в оперативной памяти, а не на внешнем устройстве.

Объявление массива должно содержать три параметра: тип каждого элемента, название массива, число элементов в массиве.

**Размер** – это константа (или константное выражение), которая определяет количество ячеек оперативной памяти, зарезервированной для массива.

Доступ к элементам массива в языке Си осуществляется двумя способами.

Первый способ – с помощью порядкового номера элемента массива, который называется индексом. В качестве индекса нельзя использовать выражение вещественного типа. Нумерация элементов массива  
начинается с 0. Индекс последнего элемента массива на единицу меньше его размерности.

Второй способ – обработка массивов с использованием указателей,  
т. к. в языке Си существует связь между массивами и указателями.

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

**Задание 1**

Ввести массив вещественных чисел размером n, n (вводится с клавиатуры). Найти его наибольший и наименьший элементы и поменять их местами. Найти сумму и произведение всех элементов массива.

#include <stdio.h>

#include <locale.h>

int main()

{

printf("Введите размер массива:\n");

int n;

if (scanf("%d", &n) != 1 || n <= 0) { // проверка на ввод натурального числа

puts("Размер массива - число натуральное!");

return 0;

}

printf("\n");

double a[1000000], sum = 0, pr = 1; // так как язык программирования С не умеет работать с динамическим объявлением массива, то мы объявляем массив с каким-то большм размером

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

for (int i = 1; i <= n; i++) {

printf("a[%d] = ", i);

if (scanf("%lf", &a[i]) != 1) { // проверка на ввод вещественного числа

puts("Элемент массива должен быть вещественным числом!");

return 0;

}

sum += a[i]; // сразу при вводе переменных, с той целью чтобы не тратить лишнее время,

pr \*= a[i]; // мы сразу в цикле посчитаем значение суммы и произведения всех эл-тов массива

}

printf("\n");

int imin = 1, imax = 1;

for (int i = 2; i <= n; i++) {

if (a[imin] > a[i]) {

imin = i;

}

if (a[imax] < a[i]) {

imax = i;

}

}

printf("Максимальный элемент - %lf, минимальный элемент - %lf\n", a[imax], a[imin]);

printf("\n");

printf("Массив до того, как мы поменяли максимальный и минимальный элементы местами:\n");

for (int i = 1; i <= n; i++) {

printf("a[%d] = %lf\n", i, a[i]);

}

printf("\n");

double temp = a[imin]; // чтобы сохранить значение наименьшего элемента массива, мы создаём временную переменную, которая и будет хранить это значение

a[imin] = a[imax]; // присваиваем элементу, где стояло минимальное значение максимальное значение

a[imax] = temp; // присваиваем элементу, где стояло максимальное значение минимальное значение в переменной temp, т.о. поменяв местами максимальное и минимальное значение

printf("Массив после того, как мы поменяли максимальный и минимальный элементы местами:\n");

for (int i = 1; i <= n; i++) {

printf("a[%d] = %lf\n", i, a[i]);

}

printf("\n");

printf("Сумма элементов массива: %lf\nПроизвдение элементов массива: %lf", sum, pr);

return 0;

}

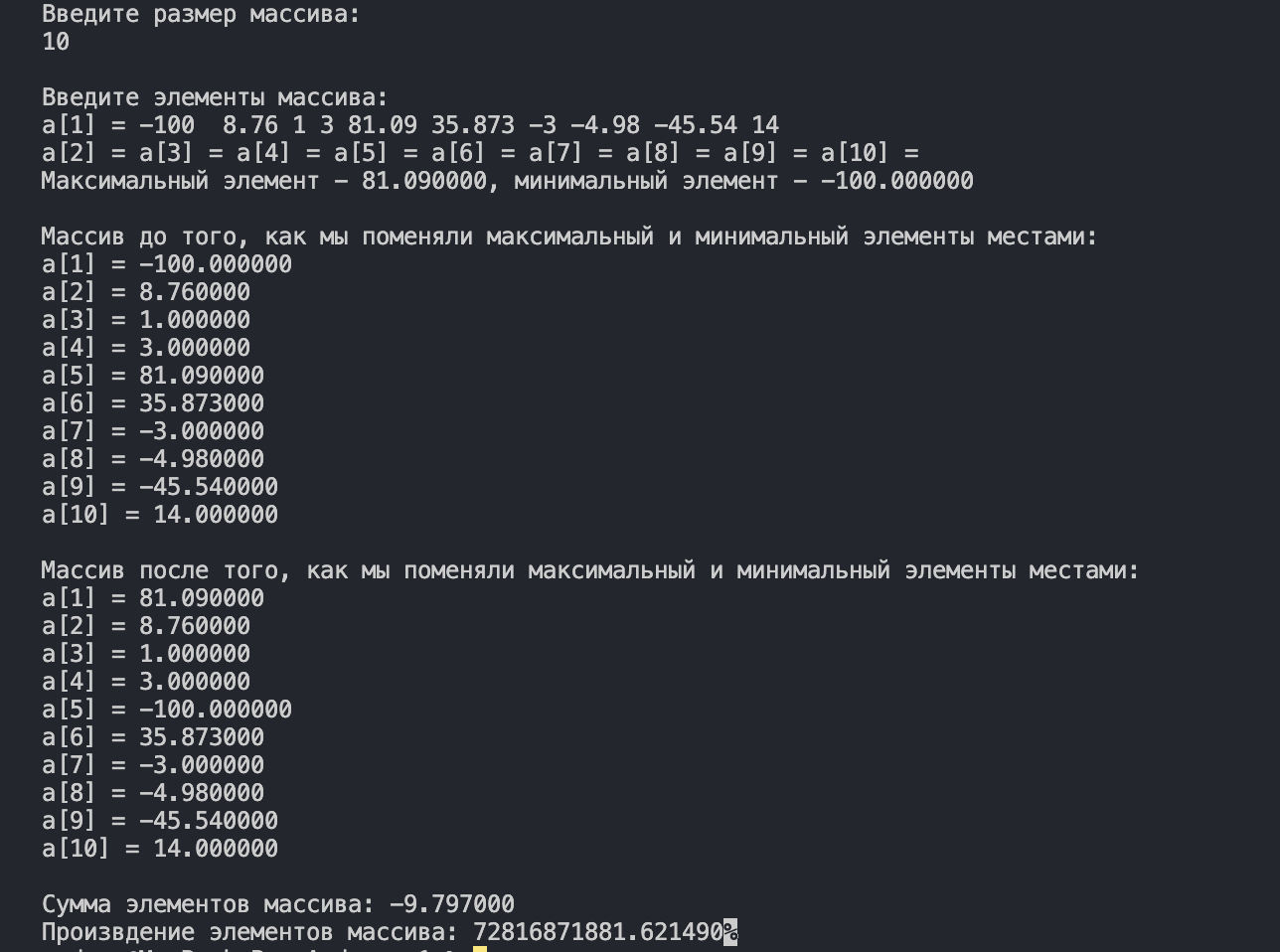


Рисунок 1.1 – Пример работы программы

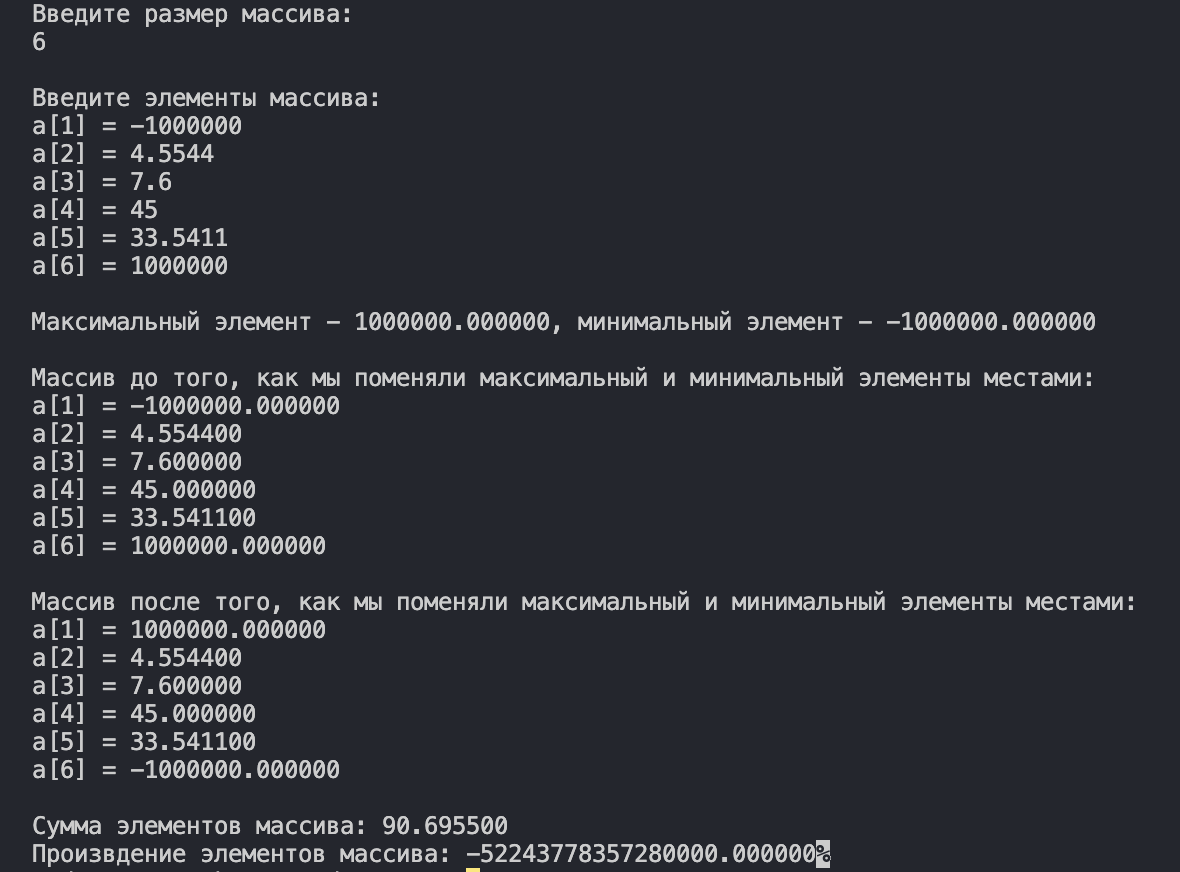


Рисунок 1.2 – Пример работы программы

**Задание 2.**

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

– произведение отрицательных элементов массива;

– сумму положительных элементов массива, расположенных до максимального элемента.

#include <stdio.h>

#include <locale.h>

int main()

{

setlocale(LC\_ALL, "rus");

float a[1000000], prNeg = 1, sumPos = 0;

int n;

printf("Введите размер массива:\n");

if (scanf("%d", &n) != 1 || n <= 0) {

puts("Размер массива должен быть числом натуральным!");

return 0;

}

printf("\n");

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

for (int i = 1; i <= n; i++) {

printf("a[%d] = ", i);

if (scanf("%f", &a[i]) != 1) {

puts("Элемент массива должен быть вещественным числом!");

return 0;

}

if (a[i] < 0) {

prNeg \*= a[i];

}

}

int imax = 1;

for (int i = 2; i <= n; i++) {

if (a[imax] < a[i]) {

imax = i;

}

}

for (int i = 1; i < imax; i++) {

if (a[i] > 0) {

sumPos += a[i];

}

}

printf("Произведение отрицательных чисел - %f,\nсумма положительных чисел, расположенных до элемента с максимальным значением - %f", prNeg, sumPos);

return 0;

}

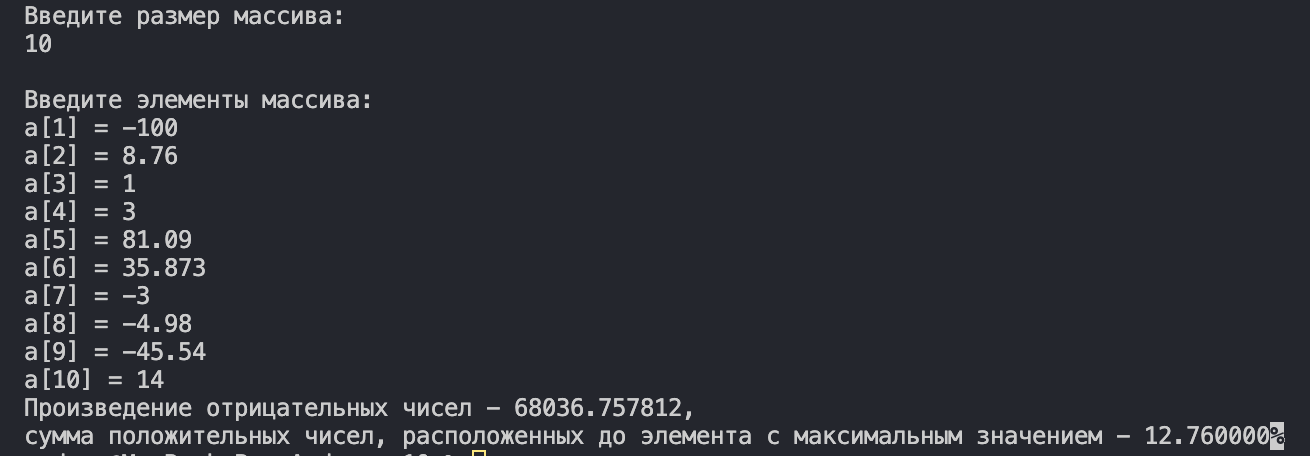
****

Рисунок 2.1 – Пример работы программы

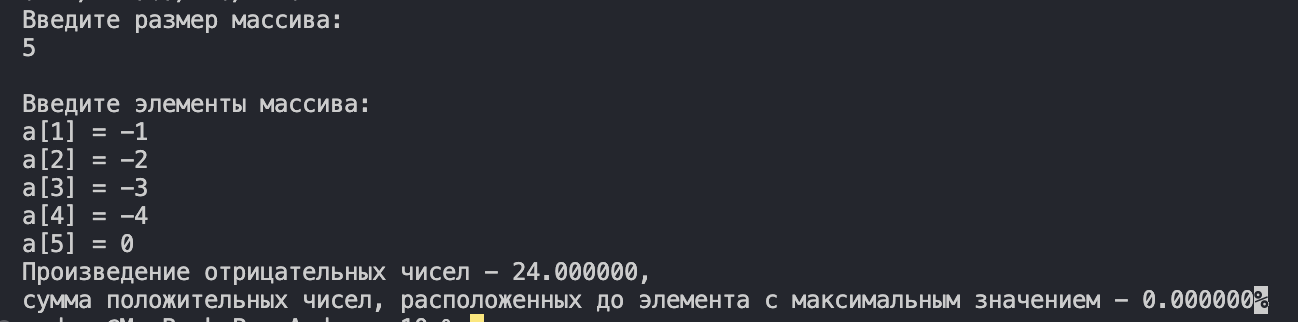


Рисунок 2.2 – Пример работы программы

**Задание 3.**

Дан массив X, содержащий 20 элементов. Все положительные элементы возвести в квадрат, а отрицательные умножить на 5. Вывести исходный и полученный массивы.

#include <stdio.h>

#include <locale.h>

int main()

{

setlocale(LC\_ALL, "rus");

double x[21];

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

for (int i = 1; i <= 20; i++) {

printf("x[%d] = ", i);

if (scanf("%lf", &x[i]) != 1) {

puts("Введённое число должно быть вещественным!");

return 0;

}

}

printf("Исходный массив:\n");

for (int i = 1; i <= 20; i++) {

printf("x[%d] = %lf\n", i, x[i]);

}

printf("\n");

for (int i = 1; i <= 20; i++) {

if (x[i] > 0) {

x[i] \*= x[i];

}

if (x[i] < 0) {

x[i] \*= 5;

}

}

printf("Полученный массив:\n");

for (int i = 1; i <= 20; i++) {

printf("x[%d] = %lf\n", i, x[i]);

}

return 0;

}

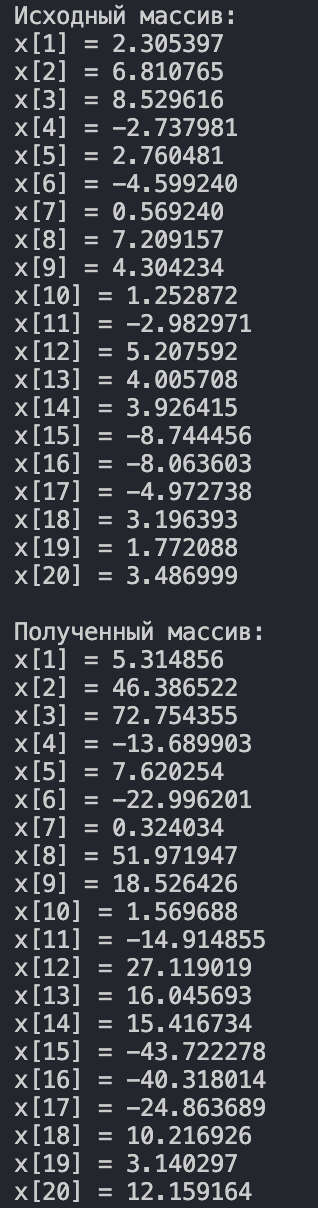


Рисунок 3.1 – Пример работы программы

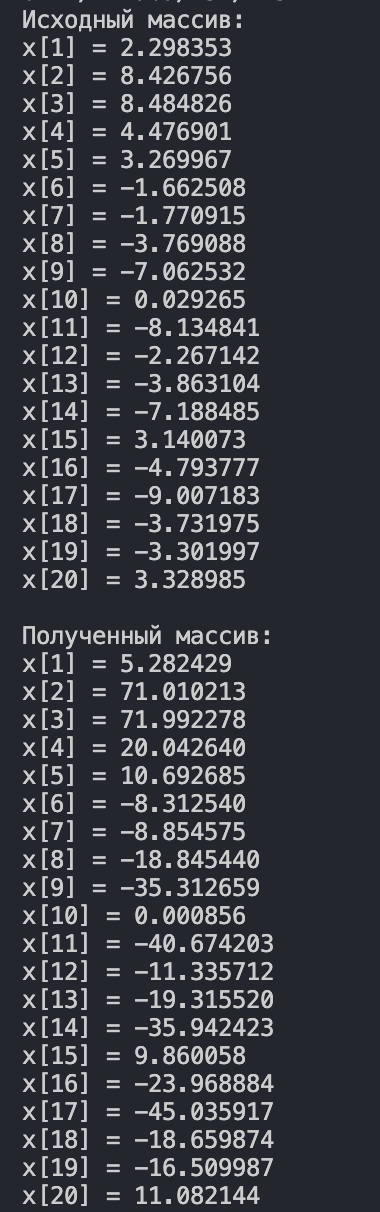


Рисунок 3.2 – Пример работы программы

ВЫВОДЫ

В ходе данной лабораторной работы я ознакомился с одномерными массивами. Узнал, что они имеют размерность, его элементы однотипны и расположены в памяти рядом. Выяснил, как получить доступ к элементу массива с помощью индекса и ссылки. Трудностей во время выполнения лабораторной работы не обнаружил.