# **Лабораторная работа № 7**

Разработка и тестирование программ на языке С# для обработки одномерных массивов.

**Цель работы:**

1. формирование навыков разработки и тестирования программ на языке С# для обработки одномерных массивов;
2. изучение основных функций для обработки одномерных массивов в языке С#
3. формирование навыков работы в среде Visual Studio.Net;

. **Оборудование:**

1. Ноутбук MSI i5
2. Программное обеспечение: ОС Windows, среда Visual Studio.Net

**Ответы на контрольные вопросы:**

1. **Каким образом определяются массивы в языке С#?**

Массивы в C# создаются с использованием оператора new и указания типа элементов и размера массива. Например: `int[] a = new int[5];`

1. **Как осуществляется доступ к отдельному элементу одномерного массива?**

Доступ к отдельному элементу массива осуществляется по индексу. Индексация начинается с нуля. Например, чтобы получить первый элемент массива, вы бы написали `a [0]`.

1. **Каким образом вводятся и выводятся одномерные массивы в языке С#?**

Ввод и вывод одномерных массивов в C# обычно осуществляется с помощью циклов. Например, можно использовать цикл for для итерации по каждому элементу массива и вывода его значения.

1. **Какие типы массивов используются в языке С#?**

В C# используются одномерные, многомерные (двумерные, трехмерные и т.д.) и зубчатые (массивы массивов) массивы.

1. **Показать, как инициализируется одномерный массив?**

Одномерный массив инициализируется следующим образом: `int[] array = new int[]{1, 2, 3, 4, 5};`

1. **Какие стандартные функции используются для обработки одномерного массива?**

Для обработки одномерного массива в C# используются различные методы класса Array, такие как Sort(), Reverse(), Length и другие. Также доступны методы из LINQ для работы с коллекциями, такие как Min(), Max(), Sum(), Average() и т.д.

**Ход работы:**

1. Разработать алгоритм задачи и представить его в виде схемы программы
2. Используя Visual Studio.Net создать файл с программой
3. Выполнить тестирование и отладку программу
4. Результаты представить в виде отчета
5. Сделать вывод о проделанной работе

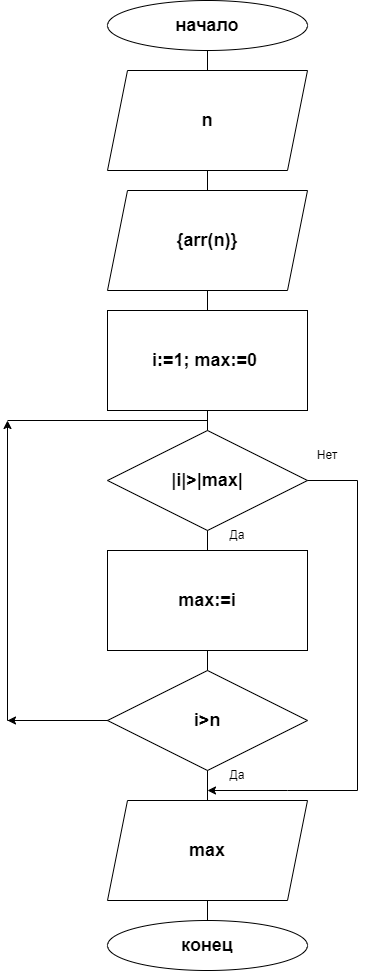
**Задание на лабораторную работу:**

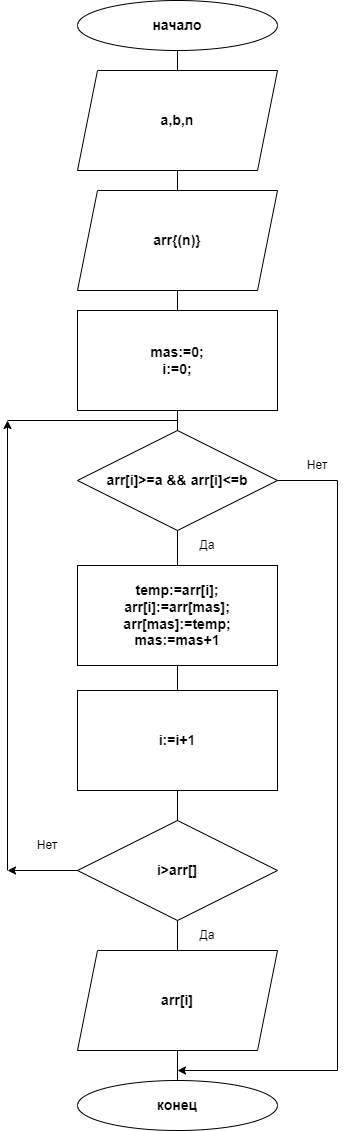
Вариант 12

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

номер максимального по модулю элемента массива. Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых лежит в интервале [а, b ], а потом — все остальные.

**Схема программы**





**Листинг с исходным кодом**

using System;

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Введите количество элементов массива:");

int n = Convert.ToInt32(Console.ReadLine());

double[] arr = new double[n];

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

{

Console.WriteLine($"Введите элемент массива[{i}]:");

arr[i] = Convert.ToDouble(Console.ReadLine());

}

int max = 0;

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

{

if (Math.Abs(arr[i]) > Math.Abs(arr[max]))

{

max = i;

}

}

Console.WriteLine($"Номер максимального по модулю элемента массива: {max}");

Console.WriteLine("Введите значение a:");

int a = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите значение b:");

int b = Convert.ToInt32(Console.ReadLine());

int mas = 0;

for (int i = 0; i < arr.Length; i++)

{

if ( (arr[i]) >= a && (arr[i]) <= b)

{

double temp = arr[i];

arr[i] = arr[mas];

arr[mas] = temp;

mas++;

}

}

Console.WriteLine("Измененный массив:");

for (int i = 0; i < arr.Length; i++)

{

Console.WriteLine(arr[i] + " ");

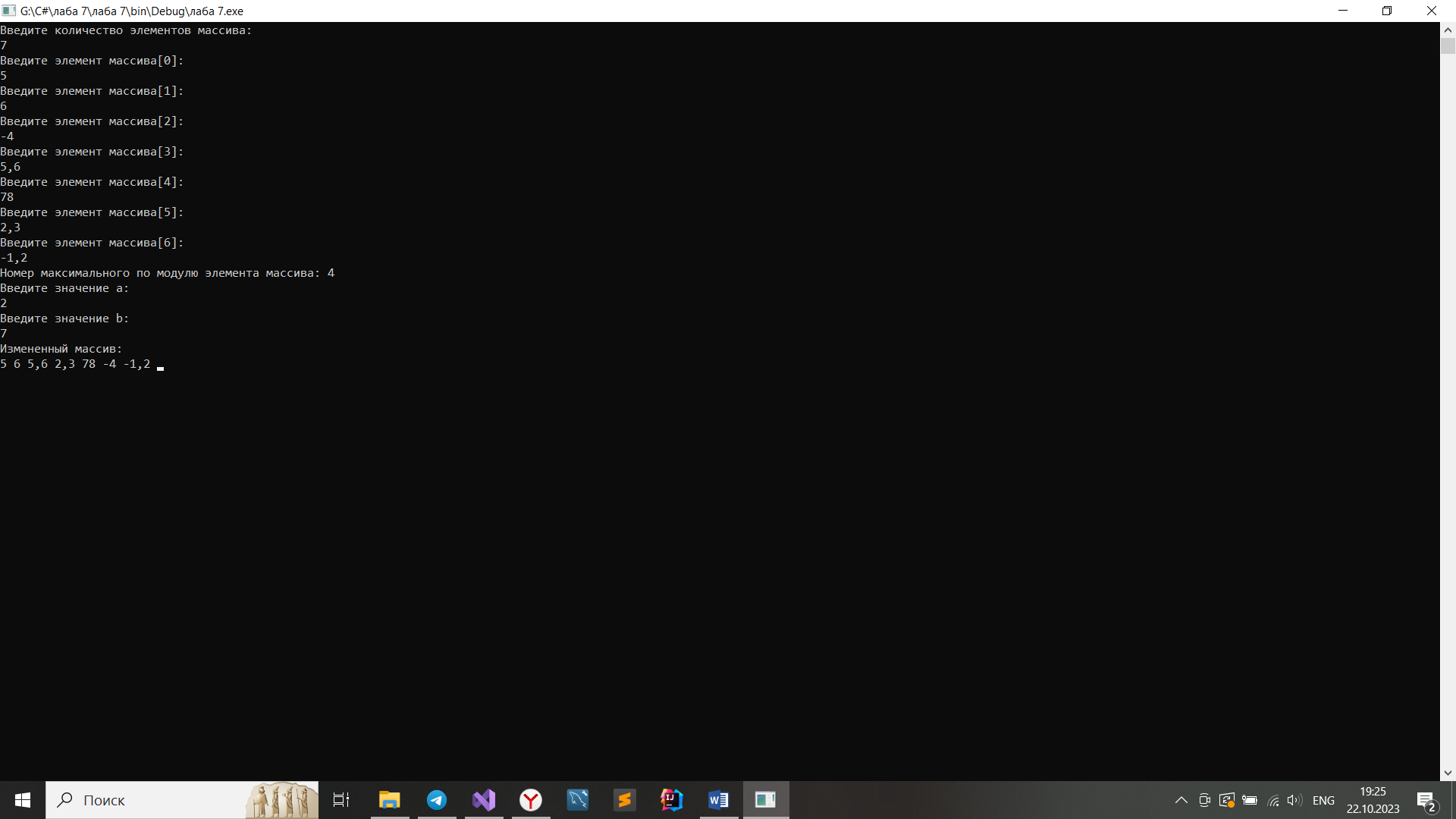
}

Console.ReadKey();

}

}

**Результаты тестирования**



**Вывод по работе**

Работа с массивами является одним из основных аспектов программирования. Массивы позволяют хранить множество значений одного типа в одной структуре данных. Они полезны во многих ситуациях, таких как обработка больших наборов данных или создание сложных структур данных.