Министерство образования Республики Беларусь

Оршанский колледж ВГУ имени П.М.Машерова

**Отчет**

по практическому заданию

«Модуль 1.2. Массивы, строки, переключатели»

по учебной практике по программированию

Выполнила учащаяся Жданова С.О.

группы 3ПОИС23 29.09.2025

Проверил Алейников М.А.

29.09.2025

Орша, 2025

**Цель:** применить на практике ранее полученные знания о массивах, строках и переключателях в языке программирования C#.

**Задания**

**Задание 1.**

Ввести размер массива N и значения его элементов. Нормировать элементы массива, разделив их на значение максимального по модулю элемента. Вывести значения элементов измененного массива.

**Задание 2.**

Определить и инициализировать целочисленный массив из 10-ти элементов. Ввести целое число и заменить им значение максимального элемента в массиве.

**Задание 3.**

Вычислить К простых чисел. Значение К ввести с клавиатуры. Вывести значения чисел, размещая их по 10 на строке.

**Задание 4.**

Определить целочисленный массив из К элементов. Присвоить элементам случайные значения из диапазона [А,В). Найти индексы минимального и максимального элементов массива. Вывести значения элементов, расположенных между найденными (включая найденные).

**Задание 5.**

Определить символьный массив из К элементов. Присвоить элементам случайные значения букв русского алфавита. Создать новый массив, поместив в него только согласные буквы из первого массива. Значение К ввести с клавиатуры. Вывести элементы обоих массивов.

**Задание 6.**

Определить вещественный массив из 10-ти элементов. Присвоить элементам случайные значения из диапазона [–10, 10). Сформировать массив индексов, которые нумеруют элементы первого массива в порядке возрастания их значений.

**Ход выполнения работы**

**Выполнение практических заданий**

**Задание 1.**

Ввести размер массива N и значения его элементов. Нормировать элементы массива, разделив их на значение максимального по модулю элемента. Вывести значения элементов измененного массива.

Ниже представлен листинг 2.1 с кодом программы для нормирования элементов массива.

Листинг 2.1. Код программы для нормирования элементов массива.

using System;

namespace ArrayNormalization

{

class Program

{

static void Main(string[] args)

{

// Выводим приветственное сообщение

Console.WriteLine("Программа нормирует массив, деля его элементы на максимальный по модулю элемент.");

// Запрашиваем размер массива

Console.Write("Введите размер массива N: ");

string inputN = Console.ReadLine(); // Считываем ввод размера

int n;

// Проверяем, является ли ввод корректным числом

if (!int.TryParse(inputN, out n) || n <= 0)

{

Console.WriteLine("Ошибка: введите положительное целое число!"); // Сообщение об ошибке

return; // Завершаем программу при некорректном вводе

}

// Создаём массив из n элементов типа double

double[] array = new double[n];

// Заполняем массив значениями от пользователя

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

{

Console.Write($"Введите элемент {i + 1}: "); // Запрашиваем элемент

string input = Console.ReadLine(); // Считываем ввод

if (!double.TryParse(input, out array[i]))

{

Console.WriteLine("Ошибка: введите корректное число!"); // Сообщение об ошибке

return; // Завершаем программу при некорректном вводе

}

}

// Выводим исходный массив

Console.WriteLine("\nИсходный массив:");

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

{

Console.Write($"{array[i]} ");

}

Console.WriteLine(); // Переходим на новую строку

// Находим максимальный по модулю элемент

double maxAbs = Math.Abs(array[0]); // Берем модуль первого элемента как начальное значение

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

{

if (Math.Abs(array[i]) > maxAbs)

{

maxAbs = Math.Abs(array[i]); // Обновляем максимум, если найден больший по модулю

}

}

// Проверяем, не равен ли максимум нулю

if (maxAbs == 0)

{

Console.WriteLine("Ошибка: все элементы массива равны нулю, деление невозможно!"); // Сообщение об ошибке

return; // Завершаем программу

}

// Нормируем массив, деля каждый элемент на максимальный по модулю

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

{

array[i] /= maxAbs; // Делим элемент на максимальный по модулю

}

// Выводим нормированный массив

Console.WriteLine("\nНормированный массив:");

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

{

Console.Write($"{array[i]:F4} "); // Форматируем до 4 знаков после запятой

}

Console.WriteLine(); // Переходим на новую строку

}

}

}

Далее запускаем написанный ранее код в среде разработке для получения результата, результат представлен на рисунке 1.2.1.

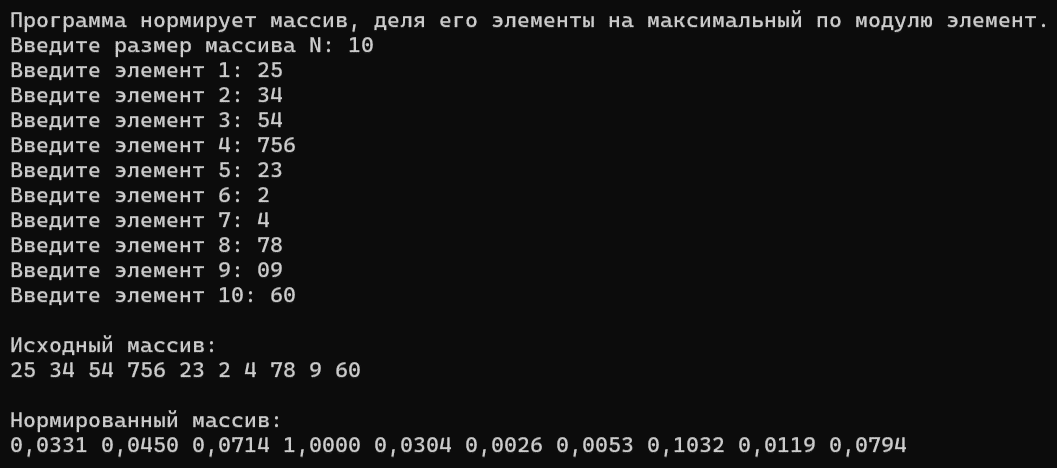


Рисунок 1.2.1 – Результат нормирования элементов массива

**Задание 2.**

Определить и инициализировать целочисленный массив из 10-ти элементов. Ввести целое число и заменить им значение максимального элемента в массиве.

Ниже представлен листинг 2.2 с кодом программы для инициализации целочисленного массива.

Листинг 2.2. Код программы для инициализации целочисленного массива.

using System;

namespace ReplaceMaxElement

{

class Program

{

static void Main(string[] args)

{

// Выводим приветственное сообщение

Console.WriteLine("Программа создаёт массив из 10 целых чисел, показывает максимальный элемент и заменяет его на введённое число.");

// Создаём объект для генерации случайных чисел

Random random = new Random();

// Создаём массив из 10 целых чисел

int[] array = new int[10];

// Заполняем массив случайными числами от 1 до 100

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

{

array[i] = random.Next(1, 101); // Генерируем число от 1 до 100 (101 не включается)

}

// Выводим исходный массив

Console.WriteLine("Исходный массив:");

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

{

Console.Write($"{array[i]} ");

}

Console.WriteLine(); // Переходим на новую строку

// Находим максимальный элемент и его индекс

int maxIndex = 0; // Предполагаем, что максимум — первый элемент

for (int i = 1; i < array.Length; i++)

{

if (array[i] > array[maxIndex])

{

maxIndex = i; // Обновляем индекс, если найден больший элемент

}

}

// Выводим максимальный элемент

Console.WriteLine($"Максимальный элемент: {array[maxIndex]}");

// Запрашиваем целое число для замены максимального элемента

Console.Write("Введите целое число для замены максимального элемента: ");

string input = Console.ReadLine(); // Считываем ввод

int newValue;

// Проверяем, является ли ввод корректным целым числом

if (!int.TryParse(input, out newValue))

{

Console.WriteLine("Ошибка: введите корректное целое число!"); // Сообщение об ошибке

return; // Завершаем программу при некорректном вводе

}

// Заменяем максимальный элемент на введённое число

array[maxIndex] = newValue; // Устанавливаем новое значение

// Выводим изменённый массив

Console.WriteLine("Изменённый массив:");

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

{

Console.Write($"{array[i]} ");

}

Console.WriteLine(); // Переходим на новую строку

}

}

}

Далее запускаем написанный ранее код в среде разработке для получения результата, результат представлен на рисунке 1.2.2.

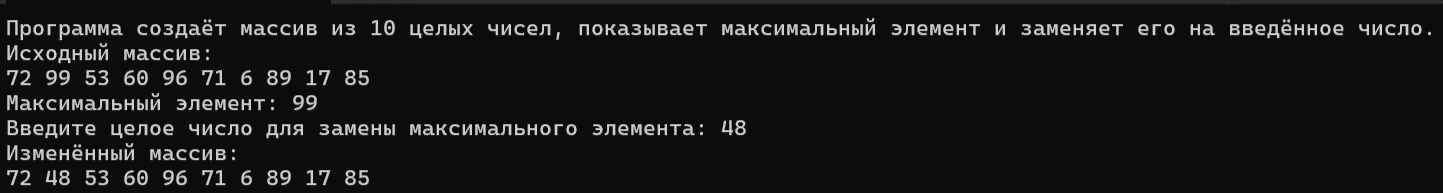


Рисунок 1.2.2 – Результат инициализации целочисленного массива

**Задание 3.**

Вычислить К простых чисел. Значение К ввести с клавиатуры. Вывести значения чисел, размещая их по 10 на строке.

Ниже представлен листинг 2.3 с кодом программы для вычисления К простых чисел и вывода значения этих чисел.

Листинг 2.3. Код программы для работы с простыми числами К.

using System;

namespace PrimeNumbers

{

class Program

{

static void Main(string[] args)

{

// Выводим приветственное сообщение

Console.WriteLine("Программа находит K простых чисел и выводит их по 10 в строке.");

// Запрашиваем количество простых чисел K

Console.Write("Введите количество простых чисел K: ");

string inputK = Console.ReadLine(); // Считываем ввод

int k;

// Проверяем, является ли ввод корректным положительным числом

if (!int.TryParse(inputK, out k) || k <= 0)

{

Console.WriteLine("Ошибка: введите положительное целое число!"); // Сообщение об ошибке

return; // Завершаем программу при некорректном вводе

}

// Создаём массив для хранения K простых чисел

int[] primes = new int[k];

// Находим K простых чисел

int count = 0; // Счётчик найденных простых чисел

int number = 2; // Начинаем проверку с 2 (первое простое число)

while (count < k)

{

if (IsPrime(number)) // Проверяем, является ли число простым

{

primes[count] = number; // Сохраняем простое число в массив

count++; // Увеличиваем счётчик

}

number++; // Переходим к следующему числу

}

// Выводим простые числа по 10 в строке

Console.WriteLine("\nПростые числа:");

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

{

Console.Write($"{primes[i],-4}"); // Выводим число с выравниванием (4 символа на число)

if ((i + 1) % 10 == 0 || i == primes.Length - 1) // Проверяем, нужно ли начать новую строку

{

Console.WriteLine(); // Переходим на новую строку после 10 чисел или в конце

}

}

}

// Метод для проверки, является ли число простым

static bool IsPrime(int number)

{

// 1 не является простым числом

if (number <= 1)

{

return false; // Возвращаем false для чисел ≤ 1

}

// Проверяем делители от 2 до квадратного корня числа

for (int i = 2; i <= Math.Sqrt(number); i++)

{

if (number % i == 0) // Если число делится на i, оно не простое

{

return false; // Возвращаем false

}

}

return true; // Число простое, если нет делителей

}

}

}

Далее запускаем написанный ранее код в среде разработке для получения результата, результат представлен на рисунке 1.2.3.

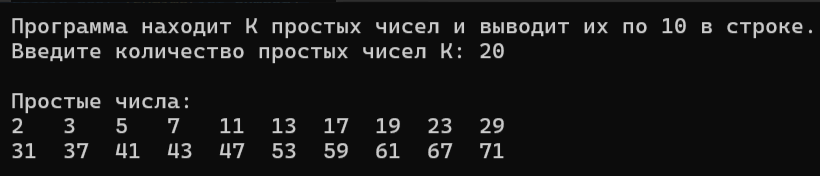


Рисунок 1.2.3 – Результат вывода чисел в консоль

**Задание 4.**

Определить целочисленный массив из К элементов. Присвоить элементам случайные значения из диапазона [А,В). Найти индексы минимального и максимального элементов массива. Вывести значения элементов, расположенных между найденными (включая найденные).

Ниже представлен листинг 2.4 с кодом программы для работы с целочисленным массивом из К элементов.

Листинг 2.4. Код программы для работы с целочисленным массивом.

using System;

namespace MinMaxIndices

{

class Program

{

static void Main(string[] args)

{

// Выводим приветственное сообщение

Console.WriteLine("Программа создаёт массив из K элементов, заполняет случайными числами из диапазона [A, B) и выводит элементы между минимальным и максимальным.");

// Запрашиваем размер массива K

Console.Write("Введите размер массива K: ");

string inputK = Console.ReadLine(); // Считываем ввод

int k;

if (!int.TryParse(inputK, out k) || k <= 0)

{

Console.WriteLine("Ошибка: введите положительное целое число для K!"); // Сообщение об ошибке

return; // Завершаем программу при некорректном вводе

}

// Запрашиваем начало диапазона A

Console.Write("Введите начало диапазона A: ");

string inputA = Console.ReadLine(); // Считываем ввод

int a;

if (!int.TryParse(inputA, out a))

{

Console.WriteLine("Ошибка: введите целое число для A!"); // Сообщение об ошибке

return; // Завершаем программу при некорректном вводе

}

// Запрашиваем конец диапазона B

Console.Write("Введите конец диапазона B: ");

string inputB = Console.ReadLine(); // Считываем ввод

int b;

if (!int.TryParse(inputB, out b) || b <= a)

{

Console.WriteLine("Ошибка: B должно быть целым числом и больше A!"); // Сообщение об ошибке

return; // Завершаем программу при некорректном вводе

}

// Создаём объект для генерации случайных чисел

Random random = new Random();

// Создаём массив из K целых чисел

int[] array = new int[k];

// Заполняем массив случайными числами из диапазона [A, B)

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

{

array[i] = random.Next(a, b); // Генерируем число от A до B-1

}

// Выводим исходный массив

Console.WriteLine("\nИсходный массив:");

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

{

Console.Write($"{array[i]} ");

}

Console.WriteLine(); // Переходим на новую строку

// Находим индексы минимального и максимального элементов

int minIndex = 0; // Предполагаем, что минимум — первый элемент

int maxIndex = 0; // Предполагаем, что максимум — первый элемент

for (int i = 1; i < array.Length; i++)

{

if (array[i] < array[minIndex])

{

minIndex = i; // Обновляем индекс минимума

}

if (array[i] > array[maxIndex])

{

maxIndex = i; // Обновляем индекс максимума

}

}

// Выводим минимальный и максимальный элементы с их индексами

Console.WriteLine($"\nМинимальный элемент: {array[minIndex]} (индекс {minIndex})");

Console.WriteLine($"Максимальный элемент: {array[maxIndex]} (индекс {maxIndex})");

// Определяем границы для вывода (от меньшего индекса к большему)

int startIndex = Math.Min(minIndex, maxIndex); // Начало диапазона

int endIndex = Math.Max(minIndex, maxIndex); // Конец диапазона

// Выводим элементы между минимальным и максимальным (включая их)

Console.WriteLine("\nЭлементы между минимальным и максимальным (включая их):");

if (startIndex == endIndex)

{

Console.WriteLine(array[startIndex]); // Если индексы совпадают, выводим только один элемент

}

else

{

for (int i = startIndex; i <= endIndex; i++)

{

Console.Write($"{array[i]} ");

}

Console.WriteLine(); // Переходим на новую строку

}

}

}

}

Далее запускаем написанный ранее код в среде разработке для получения результата, результат представлен на рисунке 1.2.4.

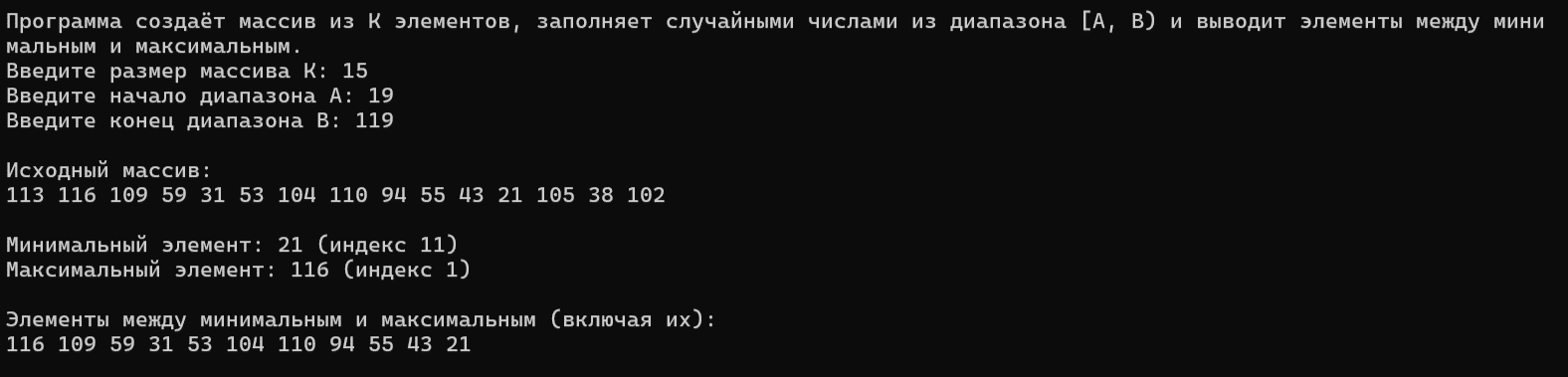


Рисунок 1.2.4 – Результат ранее написанного кода полученный в консоли

**Задание 5.**

Определить символьный массив из К элементов. Присвоить элементам случайные значения букв русского алфавита. Создать новый массив, поместив в него только согласные буквы из первого массива. Значение К ввести с клавиатуры. Вывести элементы обоих массивов.

Ниже представлен листинг 2.5 с кодом программы для работы с символьным массивом, элементы которого представляют собой значения букв русского алфавита.

Листинг 2.5. Код программы для работы с символьным массивом.

using System;

using System.Collections.Generic;

namespace RussianConsonants

{

class Program

{

static void Main(string[] args)

{

// Выводим приветственное сообщение

Console.WriteLine("Программа создаёт массив из K случайных букв русского алфавита и выделяет согласные.");

// Запрашиваем размер массива K

Console.Write("Введите размер массива K: ");

string inputK = Console.ReadLine(); // Считываем ввод

int k;

if (!int.TryParse(inputK, out k) || k <= 0)

{

Console.WriteLine("Ошибка: введите положительное целое число для K!"); // Сообщение об ошибке

return; // Завершаем программу при некорректном вводе

}

// Создаём объект для генерации случайных чисел

Random random = new Random();

// Определяем русский алфавит (33 буквы)

char[] russianAlphabet = { 'а', 'б', 'в', 'г', 'д', 'е', 'ё', 'ж', 'з', 'и', 'й',

'к', 'л', 'м', 'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф',

'х', 'ц', 'ч', 'ш', 'щ', 'ъ', 'ы', 'ь', 'э', 'ю', 'я' };

// Создаём массив из K символов

char[] letters = new char[k];

// Заполняем массив случайными буквами русского алфавита

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

{

letters[i] = russianAlphabet[random.Next(0, russianAlphabet.Length)]; // Выбираем случайную букву

}

// Выводим исходный массив

Console.WriteLine("\nИсходный массив букв:");

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

{

Console.Write($"{letters[i]} ");

}

Console.WriteLine(); // Переходим на новую строку

// Создаём список для согласных букв

List<char> consonants = new List<char>();

// Определяем гласные буквы для фильтрации

char[] vowels = { 'а', 'е', 'ё', 'и', 'о', 'у', 'ы', 'э', 'ю', 'я' };

// Фильтруем согласные буквы

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

{

bool isVowel = false;

for (int j = 0; j < vowels.Length; j++)

{

if (letters[i] == vowels[j])

{

isVowel = true; // Если буква гласная, помечаем

break;

}

}

if (!isVowel && letters[i] != 'ь' && letters[i] != 'ъ' && letters[i] != 'й') // Исключаем мягкий/твёрдый знак и 'й'

{

consonants.Add(letters[i]); // Добавляем согласную в список

}

}

// Выводим массив согласных

Console.WriteLine("\nМассив согласных букв:");

if (consonants.Count == 0)

{

Console.WriteLine("Согласные буквы отсутствуют.");

}

else

{

for (int i = 0; i < consonants.Count; i++)

{

Console.Write($"{consonants[i]} ");

}

Console.WriteLine(); // Переходим на новую строку

}

}

}

}

Далее запускаем написанный ранее код в среде разработке для получения результата, результат представлен на рисунке 1.2.5.

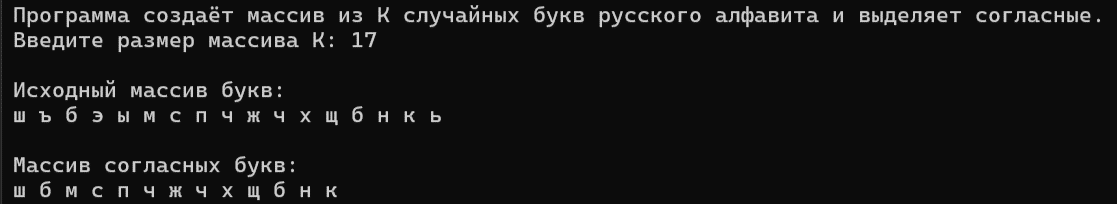


Рисунок 1.2.5 – Результат работы с символьным массивом

**Задание 6.**

Определить вещественный массив из 10-ти элементов. Присвоить элементам случайные значения из диапазона [–10, 10). Сформировать массив индексов, которые нумеруют элементы первого массива в порядке возрастания их значений.

Ниже представлен листинг 2.6 с кодом программы для работы с вещественным массивом из 10-ти элементов.

Листинг 2.6. Код программы для работы с вещественным массивом.

using System;

namespace SortedIndices

{

class Program

{

static void Main(string[] args)

{

// Выводим приветственное сообщение

Console.WriteLine("Программа создаёт массив из 10 вещественных чисел в диапазоне [-10, 10) и формирует массив индексов по возрастанию значений.");

Console.WriteLine("Результаты будут показаны поэтапно. Нажмите Enter для продолжения.");

// Ожидаем нажатия Enter для начала

Console.ReadLine();

// Создаём объект для генерации случайных чисел

Random random = new Random();

// Создаём массив из 10 вещественных чисел

double[] numbers = new double[10];

// Заполняем массив случайными числами из диапазона [-10, 10)

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

{

numbers[i] = random.NextDouble() \* 20 - 10; // Генерируем число от -10 до 10

}

// Выводим исходный массив

Console.WriteLine("\nИсходный массив:");

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

{

Console.Write($"{numbers[i]:F2} "); // Форматируем до 2 знаков после запятой

}

Console.WriteLine("\n\nНажмите Enter, чтобы найти минимальный и максимальный элементы.");

Console.ReadLine(); // Ожидаем подтверждение пользователя

// Находим минимальный и максимальный элементы

int minIndex = 0;

int maxIndex = 0;

for (int i = 1; i < numbers.Length; i++)

{

if (numbers[i] < numbers[minIndex])

{

minIndex = i; // Обновляем индекс минимума

}

if (numbers[i] > numbers[maxIndex])

{

maxIndex = i; // Обновляем индекс максимума

}

}

// Выводим минимальный и максимальный элементы

Console.WriteLine($"\nМинимальный элемент: {numbers[minIndex]:F2} (индекс {minIndex})");

Console.WriteLine($"Максимальный элемент: {numbers[maxIndex]:F2} (индекс {maxIndex})");

Console.WriteLine("\nНажмите Enter, чтобы сформировать массив индексов.");

Console.ReadLine(); // Ожидаем подтверждение пользователя

// Создаём массив индексов

int[] indices = new int[10];

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

{

indices[i] = i; // Инициализируем индексы от 0 до 9

}

// Сортируем массив индексов по значениям чисел в numbers

for (int i = 0; i < indices.Length - 1; i++)

{

for (int j = 0; j < indices.Length - 1 - i; j++)

{

if (numbers[indices[j]] > numbers[indices[j + 1]])

{

// Меняем индексы местами

int temp = indices[j];

indices[j] = indices[j + 1];

indices[j + 1] = temp;

}

}

}

// Выводим массив индексов

Console.WriteLine("\nМассив индексов (по возрастанию значений):");

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

{

Console.Write($"{indices[i]} ");

}

Console.WriteLine("\n\nНажмите Enter, чтобы показать элементы в порядке возрастания.");

Console.ReadLine(); // Ожидаем подтверждение пользователя

// Выводим элементы в порядке возрастания

Console.WriteLine("\nЭлементы массива в порядке возрастания:");

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

{

Console.Write($"{numbers[indices[i]]:F2} ");

}

Console.WriteLine("\n\nПрограмма завершена. Нажмите Enter для выхода.");

Console.ReadLine(); // Ожидаем завершение

}

}

}

Далее запускаем написанный ранее код в среде разработке для получения результата, результат представлен на рисунке 1.2.6.

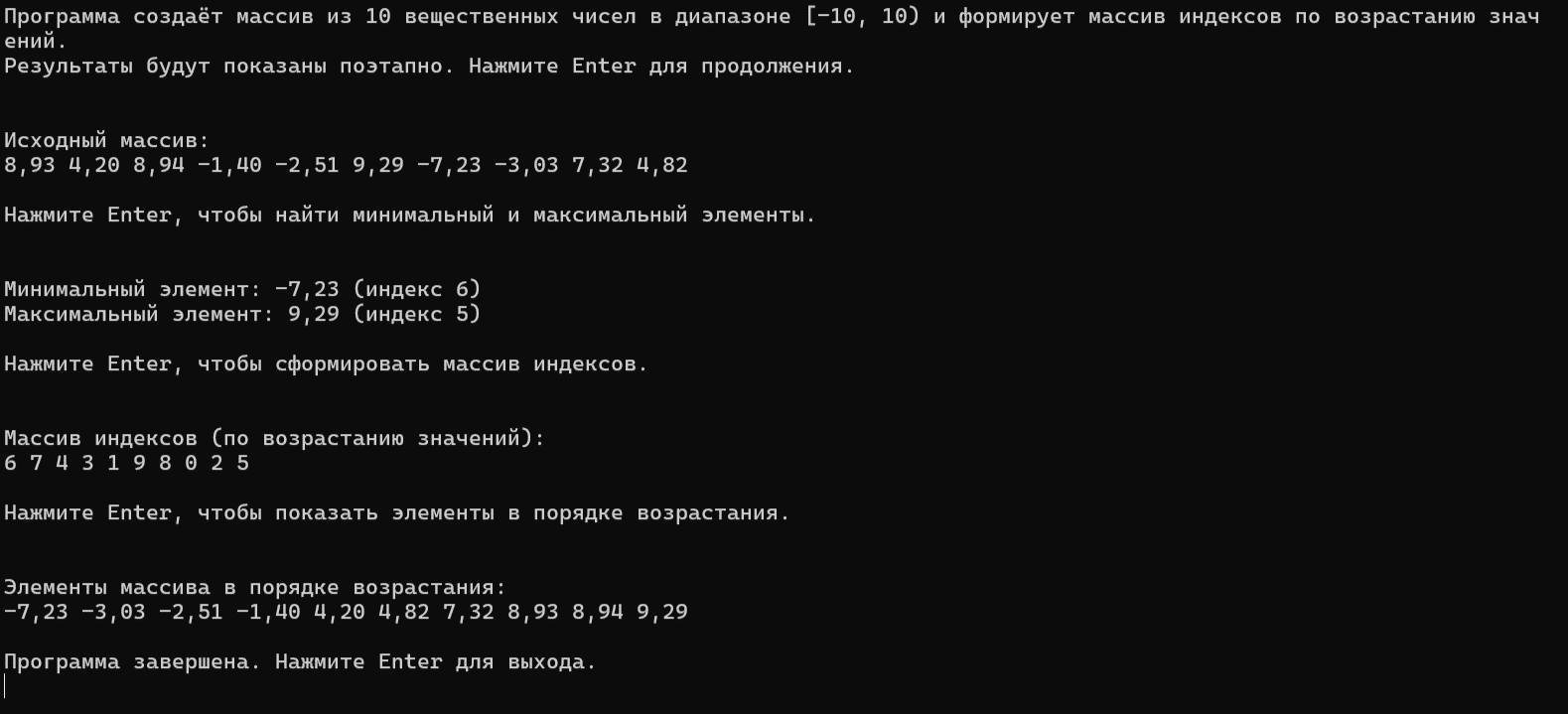


Рисунок 1.2.6 – Результат работы с целочисленным массивом

**Вывод:** в ходе выполнения практического задания были применены на практике ранее полученные знания о массивах, строках и переключателях в языке программирования C#.