**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ** **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** **«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ** **“СИНЕРГИЯ”»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | Синергия |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | Информационные системы и программирование |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

**Отчет по лабораторной работе №1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **на тему** |  | Линейные программы | | |
|  |  | (наименование темы) | | |
|  |  |  | | |
| **по дисциплине** | | |  | Архитектура аппаратных средств |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Шитик Алина Эдуардовна |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | ДКИП-206ПРОГ |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | Сибирев Иван Валерьевич |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2024 г.**

Лабораторная работа 5. Одномерные массивы

Вариант 1

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

* сумму отрицательных элементов массива;
* произведение элементов массива, расположенных между максимальным и минимальным элементами. Упорядочить элементы массива по возрастанию.

Листинг кода:

using System;

class Program

{

static void Main()

{

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

int n = int.Parse(Console.ReadLine());

double[] array = new double[n];

// Заполнение массива

Console.WriteLine($"Введите {n} вещественных элементов массива:");

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

{

array[i] = double.Parse(Console.ReadLine());

}

// 1. Сумма отрицательных элементов

double sumNegative = 0;

foreach (double num in array)

{

if (num < 0)

{

sumNegative += num;

}

}

Console.WriteLine($"Сумма отрицательных элементов: {sumNegative}");

// 2. Произведение между максимальным и минимальным элементами

if (array.Length < 2)

{

Console.WriteLine("Недостаточно элементов для вычисления произведения между max и min");

}

else

{

int maxIndex = 0, minIndex = 0;

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

{

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

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

}

int start = Math.Min(maxIndex, minIndex);

int end = Math.Max(maxIndex, minIndex);

double product = 1;

bool hasElements = false;

for (int i = start + 1; i < end; i++)

{

product \*= array[i];

hasElements = true;

}

if (hasElements)

{

Console.WriteLine($"Произведение элементов между max и min: {product}");

}

else

{

Console.WriteLine("Между максимальным и минимальным элементами нет других элементов");

}

}

// Сортировка массива по возрастанию

Array.Sort(array);

Console.WriteLine("Отсортированный массив по возрастанию:");

foreach (double num in array)

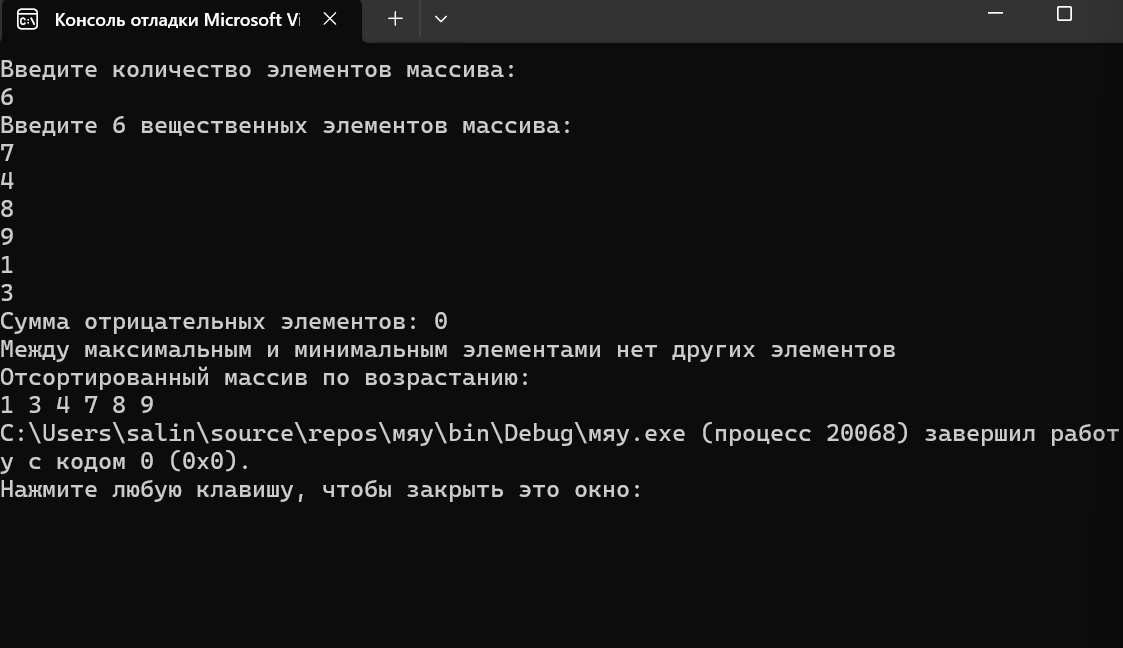
{

Console.Write(num + " ");

}

}

}



Лабораторная работа 6. Двумерные массивы

Листинг кода

using System;

using System.Linq;

class Program

{

static void Main()

{

// Ввод размеров матрицы

Console.WriteLine("Введите количество строк матрицы:");

int rows = int.Parse(Console.ReadLine());

Console.WriteLine("Введите количество столбцов матрицы:");

int cols = int.Parse(Console.ReadLine());

// Создание и заполнение матрицы

int[,] matrix = new int[rows, cols];

Console.WriteLine($"Введите {rows \* cols} элементов матрицы:");

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

{

for (int j = 0; j < cols; j++)

{

Console.Write($"Элемент [{i},{j}]: ");

matrix[i, j] = int.Parse(Console.ReadLine());

}

}

// Вывод исходной матрицы

Console.WriteLine("\nИсходная матрица:");

PrintMatrix(matrix);

// 1. Количество строк без нулевых элементов

int nonZeroRows = CountRowsWithoutZeros(matrix);

Console.WriteLine($"\n1. Количество строк без нулевых элементов: {nonZeroRows}");

// 2. Максимальное число, встречающееся более одного раза

int maxRepeatingNumber = FindMaxRepeatingNumber(matrix);

if (maxRepeatingNumber != int.MinValue)

{

Console.WriteLine($"2. Максимальное число, встречающееся более одного раза: {maxRepeatingNumber}");

}

else

{

Console.WriteLine("2. В матрице нет чисел, встречающихся более одного раза");

}

}

// Метод для вывода матрицы

static void PrintMatrix(int[,] matrix)

{

int rows = matrix.GetLength(0);

int cols = matrix.GetLength(1);

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

{

for (int j = 0; j < cols; j++)

{

Console.Write($"{matrix[i, j],4} ");

}

Console.WriteLine();

}

}

// Метод для подсчета строк без нулей

static int CountRowsWithoutZeros(int[,] matrix)

{

int count = 0;

int rows = matrix.GetLength(0);

int cols = matrix.GetLength(1);

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

{

bool hasZero = false;

for (int j = 0; j < cols; j++)

{

if (matrix[i, j] == 0)

{

hasZero = true;

break;

}

}

if (!hasZero)

{

count++;

}

}

return count;

}

// Метод для поиска максимального повторяющегося числа

static int FindMaxRepeatingNumber(int[,] matrix)

{

int rows = matrix.GetLength(0);

int cols = matrix.GetLength(1);

int maxNumber = int.MinValue;

// Создаем словарь для подсчета количества вхождений каждого числа

var numberCounts = new System.Collections.Generic.Dictionary<int, int>();

// Подсчитываем количество вхождений каждого числа

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

{

for (int j = 0; j < cols; j++)

{

int currentNumber = matrix[i, j];

if (numberCounts.ContainsKey(currentNumber))

{

numberCounts[currentNumber]++;

}

else

{

numberCounts.Add(currentNumber, 1);

}

}

}

// Находим максимальное число, встречающееся более одного раза

foreach (var pair in numberCounts)

{

if (pair.Value > 1 && pair.Key > maxNumber)

{

maxNumber = pair.Key;

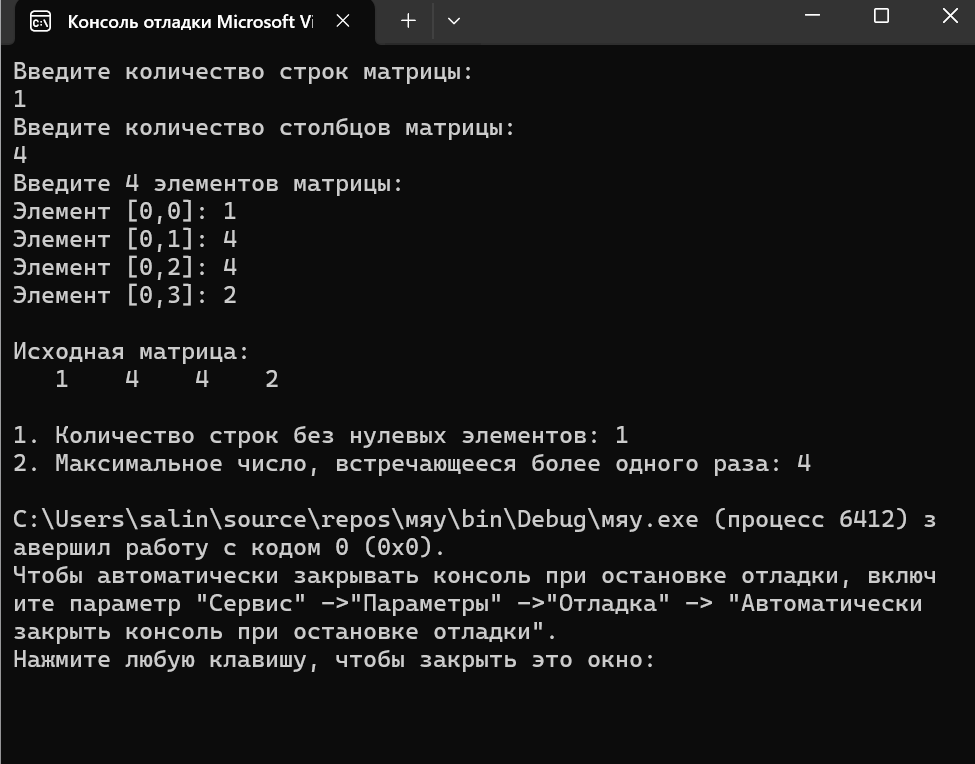
}

}

return maxNumber;

}

}



Лабораторная работа 7. Строки

Вариант 1 Написать программу, которая считывает из текстового файла три предложения и выводит их в обратном порядке.

Листинг кода:

using System;

using System.IO;

using System.Text;

class Program

{

static void Main()

{

Console.WriteLine("Программа для чтения трех предложений из файла и вывода их в обратном порядке");

try

{

// 1. Запрос пути к файлу у пользователя

Console.WriteLine("\nВведите полный путь к текстовому файлу:");

string filePath = Console.ReadLine();

// 2. Чтение всего содержимого файла

string fileContent = File.ReadAllText(filePath, Encoding.UTF8);

// 3. Разделение текста на предложения

string[] sentences = SplitIntoSentences(fileContent);

// 4. Проверка наличия минимум 3 предложений

if (sentences.Length < 3)

{

Console.WriteLine($"\nОшибка: файл содержит только {sentences.Length} предложений. Требуется минимум 3.");

return;

}

// 5. Вывод первых трех предложений в обратном порядке

Console.WriteLine("\nПервые три предложения в обратном порядке:");

for (int i = 2; i >= 0; i--)

{

Console.WriteLine($"{3 - i}. {sentences[i].Trim()}");

}

}

catch (FileNotFoundException)

{

Console.WriteLine("\nОшибка: файл не найден. Проверьте путь и повторите попытку.");

}

catch (Exception ex)

{

Console.WriteLine($"\nПроизошла ошибка: {ex.Message}");

}

}

// Метод для разделения текста на предложения

static string[] SplitIntoSentences(string text)

{

// Упрощенное разделение по точкам, восклицательным и вопросительным знакам

char[] separators = { '.', '!', '?' };

// Разделение с учетом пустых записей и удалением пустых элементов

string[] sentences = text.Split(separators, StringSplitOptions.RemoveEmptyEntries);

// Очистка предложений от лишних пробелов

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

{

sentences[i] = sentences[i].Trim();

}

return sentences;

}

}

