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

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

**Отчет по лабораторной работе №4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **на тему** |  | Массивы и строки | | |
|  |  | (наименование темы) | | |
|  |  |  | | |
| **по дисциплине** | | |  | Разработка программных модулей |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Осовицкий Б.Н. |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | ДКИП-205прог |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | Сиберев И.В. |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2025 г.**

**Лабораторная работа №4. «Массивы и строки»**

**Задание №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()

    {

        double[] array = { 3.2, 4.5, 1.7, 0.8, 2.3, 5.6, 0.9, 3.1 };

        Console.WriteLine("Исходный массив:");

        PrintArray(array);

        // 1. Сумма отрицательных элементов

        double sumNeg = 0;

        foreach (double num in array)

        {

            if (num < 0) sumNeg += num;

        }

        Console.WriteLine($"\nСумма отрицательных: {sumNeg:F2}");

        // 2. Произведение между max и min

        if (array.Length > 1)

        {

            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) + 1;

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

            double product = 1.0;

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

            {

                product \*= array[i];

            }

            Console.WriteLine($"Произведение между max и min: {product:F2}");

        }

        // 3. Сортировка

        Array.Sort(array);

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

        PrintArray(array);

    }

    static void PrintArray(double[] arr)

    {

        foreach (double num in arr)

        {

            Console.Write($"{num:F2} ");

        }

        Console.WriteLine();

    }

}

**Код на компиляторе:**

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, Шрифт, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

**Задание №2.**

Дана пелочисленная прямоугольная матрица.

Определить:

количество строк, не содержащих ни одного нулевого элемента;

максимальное из чисел, встречающихся в заданной матрице более одного раза.

**Листинг кода:**

using System;

using System.Collections.Generic;

class Program

{

static void Main()

{

Console.Write("Введите количество строк: ");

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

Console.Write("Введите количество столбцов: ");

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

int[,] matrix = new int[m, n];

// Ввод матрицы

Console.WriteLine("Введите элементы матрицы:");

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

{

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

{

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

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

}

}

// 1. Кол-во строк без нулей

int nonZeroRows = 0;

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

{

bool hasZero = false;

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

{

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

{

hasZero = true;

break;

}

}

if (!hasZero)

{

nonZeroRows++;

}

}

// 2. Максимум из повторяющихся элементов

Dictionary<int, int> counts = new Dictionary<int, int>();

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

{

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

{

int value = matrix[i, j];

if (counts.ContainsKey(value))

counts[value]++;

else

counts[value] = 1;

}

}

int maxDuplicate = int.MinValue;

bool foundDuplicate = false;

foreach (var pair in counts)

{

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

{

maxDuplicate = pair.Key;

foundDuplicate = true;

}

}

// Результаты

Console.WriteLine($"\nКоличество строк без нулей: {nonZeroRows}");

if (foundDuplicate)

Console.WriteLine($"Максимальный повторяющийся элемент: {maxDuplicate}");

else

Console.WriteLine("Нет повторяющихся элементов.");

}

}

Код на компиляторе:

Изображение выглядит как текст, снимок экрана, меню

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

**Задание 3.**

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

**Листинг кода:**

using System;

using System.IO;

class Program

{

static void Main()

{

string filePath = "text.txt";

if (!File.Exists(filePath))

{

Console.WriteLine("Файл не найден.");

return;

}

// Чтение содержимого файла

string content = File.ReadAllText(filePath);

// Разделение на предложения по стандартным окончаниям

string[] sentences = content.Split(new char[] { '.', '!', '?' }, StringSplitOptions.RemoveEmptyEntries);

// Обрезаем пробелы

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

{

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

}

// Проверка на наличие хотя бы трёх предложений

if (sentences.Length < 3)

{

Console.WriteLine("В файле меньше трёх предложений.");

return;

}

// Вывод последних трёх предложений в обратном порядке

Console.WriteLine("\nПредложения в обратном порядке:\n");

for (int i = 0; i < 3; i++)

{

Console.WriteLine(sentences[sentences.Length - 1 - i]);

}

}

}

Код на компиляторе:

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.