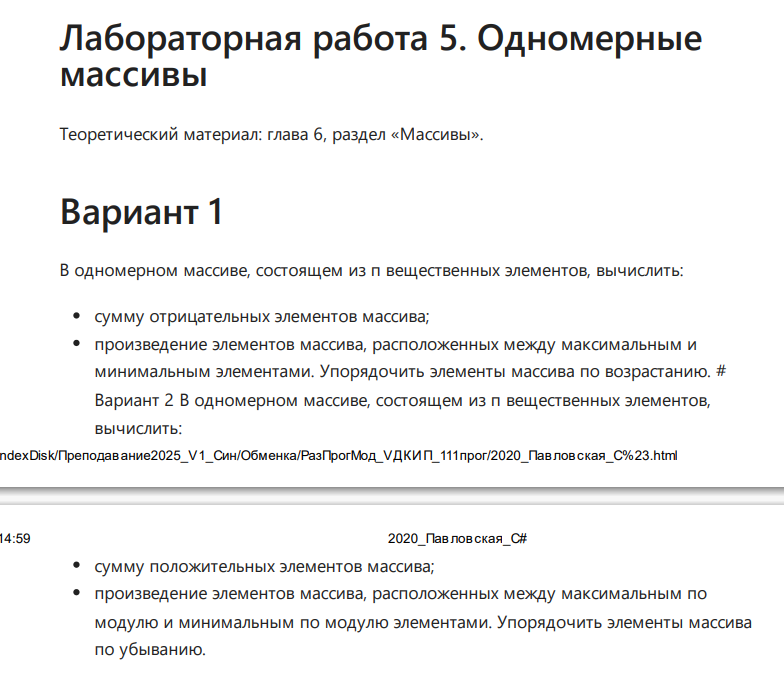
Албул Елизар Максимович ДКИП-201прог



using System;

using System.Linq;

class Program

{

static void Main()

{

Console.WriteLine("Вариант 1. Обработка одномерного массива");

// Ввод массива

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

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

double[] array = new double[n];

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

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

{

Console.Write($"Элемент {i + 1}: ");

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

}

// 1. Сумма отрицательных элементов

double sumNegatives = array.Where(x => x < 0).Sum();

Console.WriteLine($"Сумма отрицательных элементов: {sumNegatives:F2}");

// 2. Произведение между max и min элементами

int maxIndex = Array.IndexOf(array, array.Max());

int minIndex = Array.IndexOf(array, array.Min());

// Определяем границы

int start = Math.Min(maxIndex, minIndex) + 1;

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

double product = 1;

if (end - start > 0)

{

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

{

product \*= array[i];

}

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

}

else

{

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

}

// 3. Сортировка по возрастанию

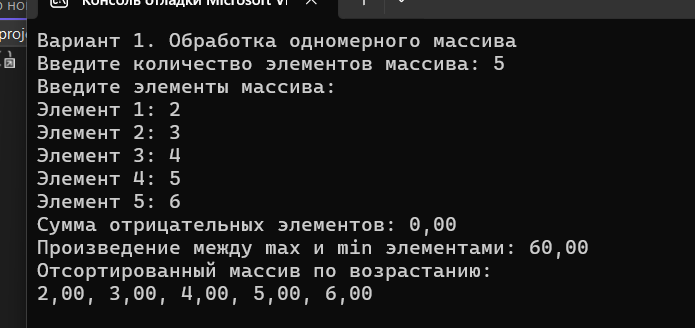
Array.Sort(array);

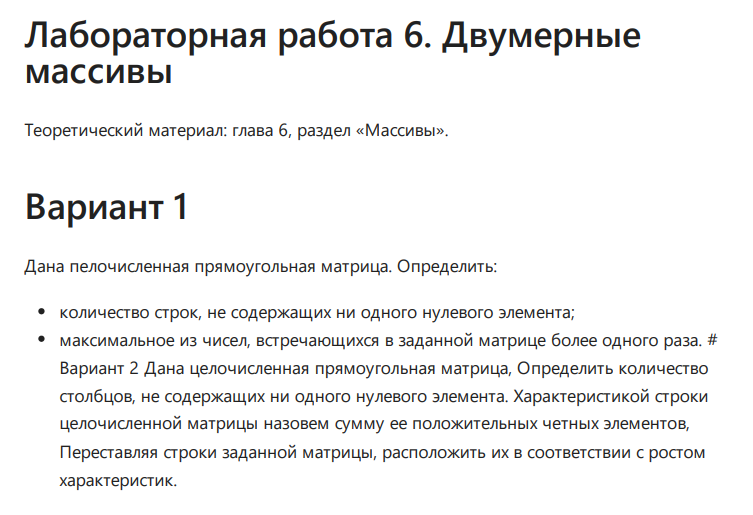
Console.WriteLine("Отсортированный массив по возрастанию:");

Console.WriteLine(string.Join(", ", array.Select(x => x.ToString("F2"))));

}

}





using System;

using System.Collections.Generic;

class Program

{

static void Main()

{

// Пример целочисленной прямоугольной матрицы

int[,] matrix = {

{1, 2, 3, 0},

{4, 5, 6, 7},

{8, 9, 0, 10},

{11, 12, 13, 14}

};

// Задача 1: Количество строк, не содержащих ни одного нулевого элемента

int countNonZeroRows = CountNonZeroRows(matrix);

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

// Задача 2: Максимальное из чисел, встречающихся более одного раза

int maxRepeatedNumber = FindMaxRepeatedNumber(matrix);

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

}

// Метод для подсчета строк без нулевых элементов

static int CountNonZeroRows(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 нахождения максимального числа, встречающегося более одного раза

static int FindMaxRepeatedNumber(int[,] matrix)

{

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

int rows = matrix.GetLength(0);

int cols = matrix.GetLength(1);

// Подсчет частоты каждого числа

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

{

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

{

int num = matrix[i, j];

if (frequency.ContainsKey(num))

{

frequency[num]++;

}

else

{

frequency[num] = 1;

}

}

}

// Поиск максимального числа, встречающегося более одного раза

int maxRepeated = int.MinValue;

foreach (var pair in frequency)

{

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

{

maxRepeated = pair.Key;

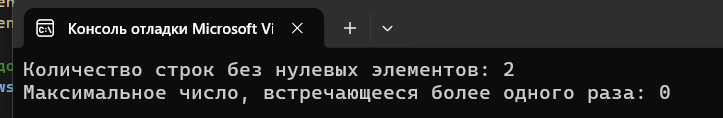
}

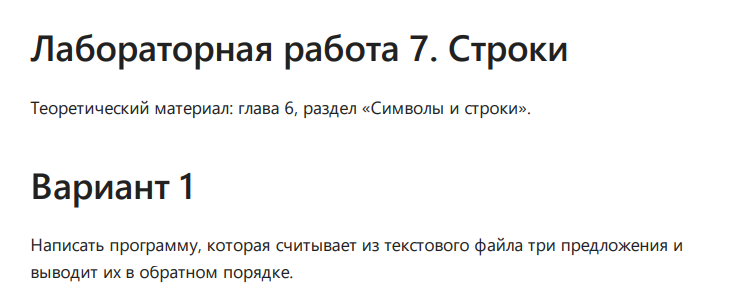
}

return maxRepeated;

}

}





using System;

using System.IO;

class Program

{

static void Main()

{

// Путь к текстовому файлу

string filePath = @"C:\Users\mrlon\OneDrive\Рабочий стол\Текстовый документ.txt";

// Чтение трех предложений из файла

string[] sentences = File.ReadAllLines(filePath);

// Вывод предложений в обратном порядке

for (int i = sentences.Length - 1; i >= 0; i--)

{

Console.WriteLine(sentences[i]);

}

}

}

