

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

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | информационных технологий |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | 09.02.07 Информационные системы и программирование |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

.

**Отчет по лабораторной работе № 3**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Бактияров Белек Бактиярович |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | Дкип111 прог |  |
|  |  |  |  |

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

**Москва, 2025**

# Лабораторная работа № 3

Оглавление

[Лабораторная работа № 3 2](#_Toc198807976)

[1.Указать свою задание с вариантом 2](#_Toc198807977)

[Задание №1 2](#_Toc198807978)

[Листинг кода 2](#_Toc198807979)

[Скрин кода 3](#_Toc198807980)

[Задание №2 4](#_Toc198807981)

[Листинг кода 4](#_Toc198807982)

[Скрин кода 5](#_Toc198807983)

[Листинг кода 5](#_Toc198807984)

[Скрин кода 7](#_Toc198807985)

[Скрин кода 7](#_Toc198807986)

# 1.Указать свою задание с вариантом

Вариант 2

В упорядоченном массиве целых чисел ai (i = 1, ..., n) найти номер находящегося в массиве элемента c, используя метод двоичного поиска.

# Задание №1

# Листинг кода

**using System;**

**using System.Linq;**

**class Program**

**{**

**static void Main()**

**{**

**double[] array = { -3.5, 2.0, -1.2, 5.8, -7.9, 0.0, 4.3 };**

**// 1. Сумма положительных элементов**

**double sumPositive = array.Where(x => x > 0).Sum();**

**Console.WriteLine("Сумма положительных элементов: " + sumPositive);**

**// 2. Произведение между макс. и мин. по модулю**

**int indexMax = Array.IndexOf(array, array.OrderByDescending(x => Math.Abs(x)).First());**

**int indexMin = Array.IndexOf(array, array.OrderBy(x => Math.Abs(x)).First());**

**int start = Math.Min(indexMax, indexMin) + 1;**

**int end = Math.Max(indexMax, indexMin);**

**double product = 1;**

**if (end - start > 0)**

**{**

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

**{**

**product \*= array[i];**

**}**

**Console.WriteLine("Произведение между макс. и мин. по модулю: " + product);**

**}**

**else**

**{**

**Console.WriteLine("Нет элементов между макс. и мин. по модулю.");**

**}**

**// 3. Упорядочивание массива по убыванию**

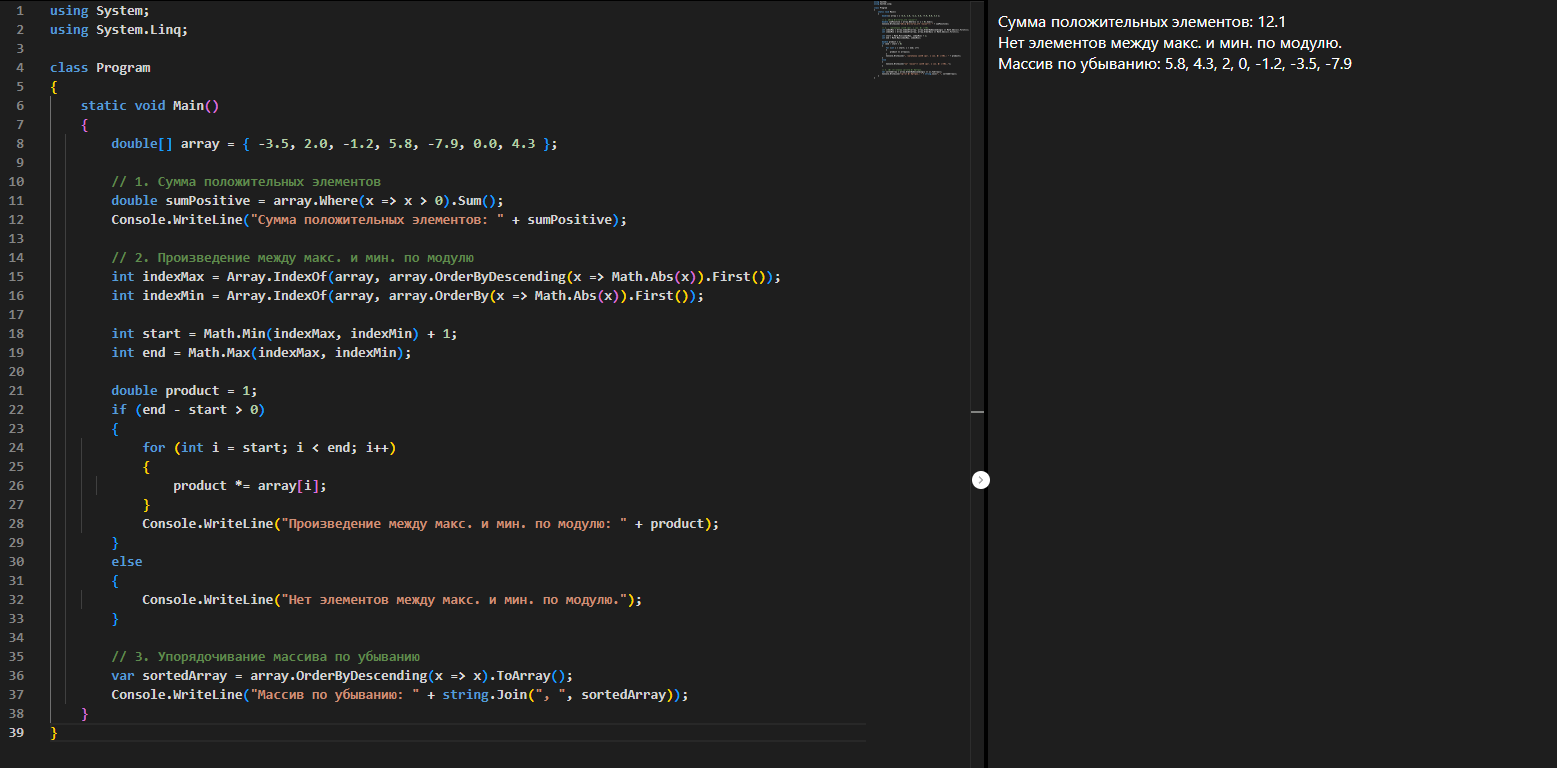
**var sortedArray = array.OrderByDescending(x => x).ToArray();**

**Console.WriteLine("Массив по убыванию: " + string.Join(", ", sortedArray));**

**}**

**}**

# Скрин кода



# Задание №2

# Листинг кода

**using System;**

**class Program**

**{**

**static void Main()**

**{**

**// Пример матрицы**

**int[,] matrix = {**

**{ 1, 0, 3 },**

**{ 4, 5, 6 },**

**{ 7, 8, 0 }**

**};**

**int rows = matrix.GetLength(0);**

**int cols = matrix.GetLength(1);**

**int count = 0;**

**for (int col = 0; col < cols; col++)**

**{**

**bool hasZero = false;**

**for (int row = 0; row < rows; row++)**

**{**

**if (matrix[row, col] == 0)**

**{**

**hasZero = true;**

**break;**

**}**

**}**

**if (!hasZero)**

**{**

**count++;**

**}**

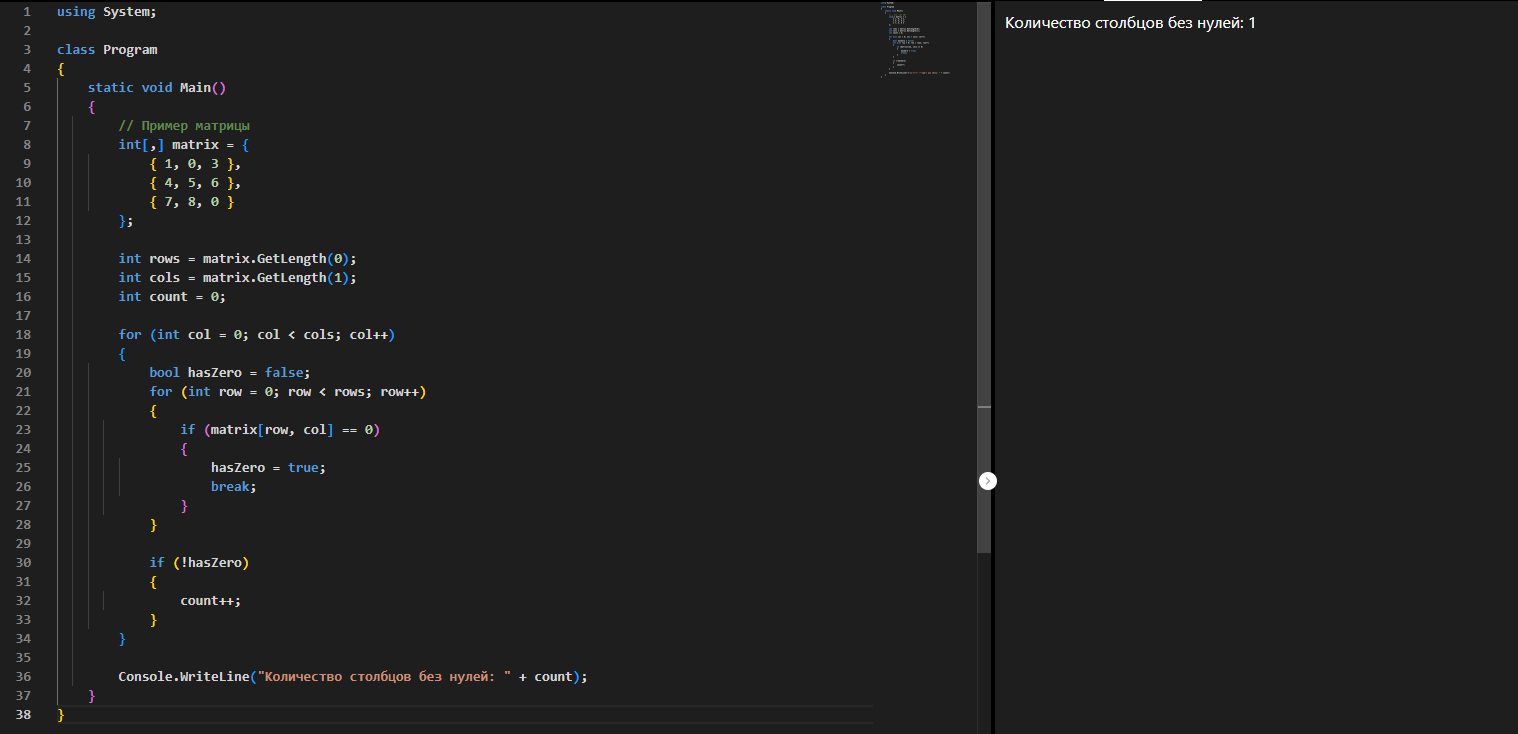
**}**

**Console.WriteLine("Количество столбцов без нулей: " + count);**

**}**

**}**

# Скрин кода



# Листинг кода

**using System;**

**class Program**

**{**

**static void Main()**

**{**

**int[,] matrix = {**

**{ 2, -4, 6 },**

**{ 1, 3, 5 },**

**{ 8, 2, -6 },**

**{ 0, 4, 7 }**

**};**

**int rows = matrix.GetLength(0);**

**int cols = matrix.GetLength(1);**

**// Храним строки и их характеристики**

**int[][] matrixRows = new int[rows][];**

**int[] characteristics = new int[rows];**

**// Заполняем массив строк и считаем характеристики**

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

**{**

**matrixRows[i] = new int[cols];**

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

**{**

**matrixRows[i][j] = matrix[i, j];**

**if (matrix[i, j] > 0 && matrix[i, j] % 2 == 0)**

**characteristics[i] += matrix[i, j];**

**}**

**}**

**// Сортировка пузырьком по характеристикам**

**for (int i = 0; i < rows - 1; i++)**

**{**

**for (int j = 0; j < rows - i - 1; j++)**

**{**

**if (characteristics[j] > characteristics[j + 1])**

**{**

**// Меняем характеристики**

**int tempChar = characteristics[j];**

**characteristics[j] = characteristics[j + 1];**

**characteristics[j + 1] = tempChar;**

**// Меняем строки местами**

**var tempRow = matrixRows[j];**

**matrixRows[j] = matrixRows[j + 1];**

**matrixRows[j + 1] = tempRow;**

**}**

**}**

**}**

**// Вывод отсортированной матрицы**

**Console.WriteLine("Матрица после сортировки строк по характеристике:");**

**foreach (var row in matrixRows)**

**{**

**foreach (var item in row)**

**{**

**Console.Write(item + "\t");**

**}**

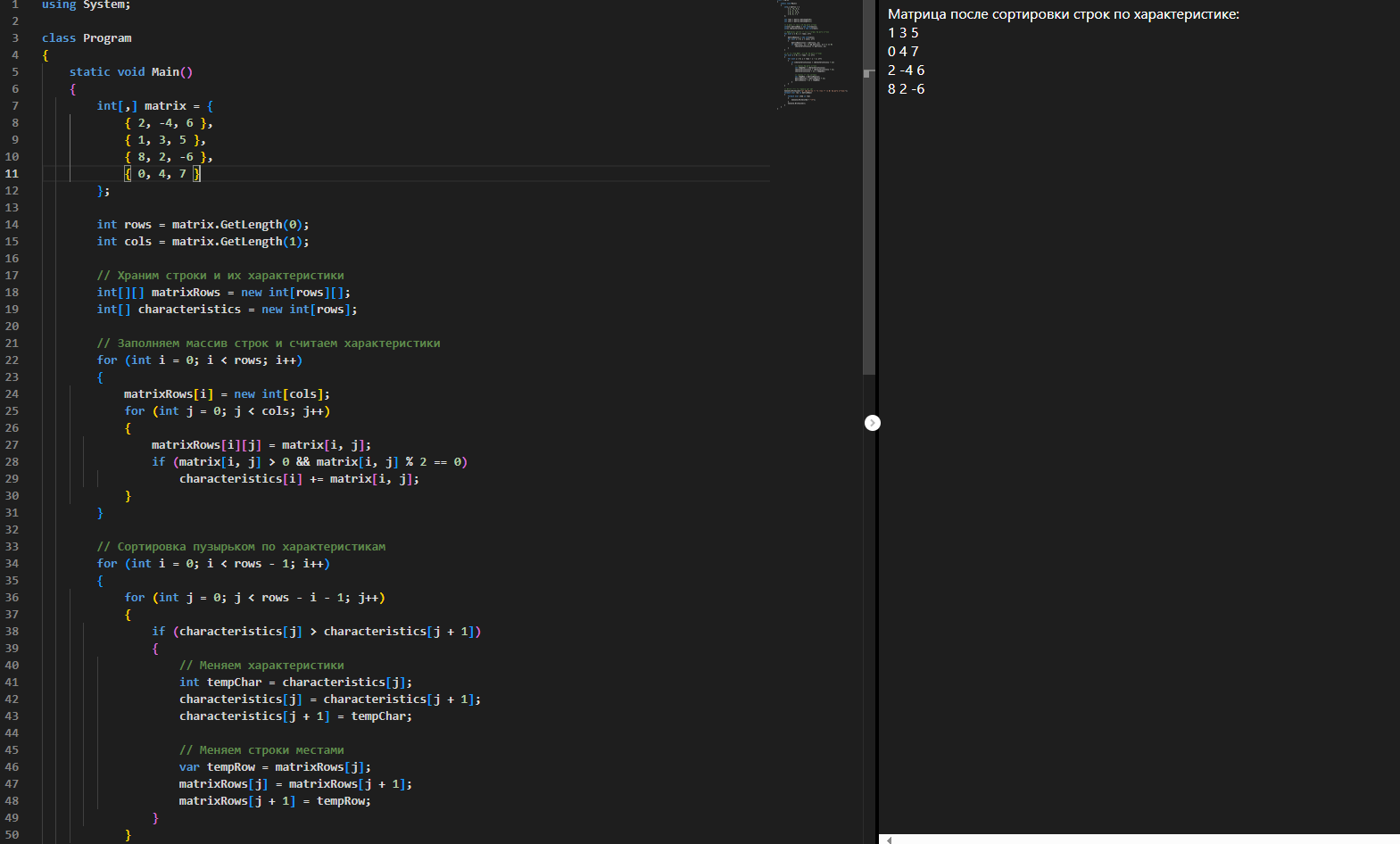
**Console.WriteLine();**

**}**

**}**

**}**

# Скрин кода



# Задание №3

# Листинг кода

**using System;**

**using System.IO;**

**using System.Text.RegularExpressions;**

**class Program**

**{**

**static void Main()**

**{**

**Console.Write("Введите путь к файлу: ");**

**string filePath = Console.ReadLine();**

**if (!File.Exists(filePath))**

**{**

**Console.WriteLine("Файл не найден.");**

**return;**

**}**

**Console.Write("Введите слово для поиска: ");**

**string keyword = Console.ReadLine();**

**string text = File.ReadAllText(filePath);**

**// Разделение текста на предложения**

**string[] sentences = Regex.Split(text, @"(?<=[\.!\?])\s+");**

**Console.WriteLine("\nПредложения, содержащие слово \"{0}\":\n", keyword);**

**foreach (string sentence in sentences)**

**{**

**// Проверка, содержит ли предложение заданное слово (целое, нечувствительное к регистру)**

**if (Regex.IsMatch(sentence, $@"\b{Regex.Escape(keyword)}\b", RegexOptions.IgnoreCase))**

**{**

**Console.WriteLine(sentence.Trim());**

**}**

**}**

**}**

**}**

# Скрин кода

