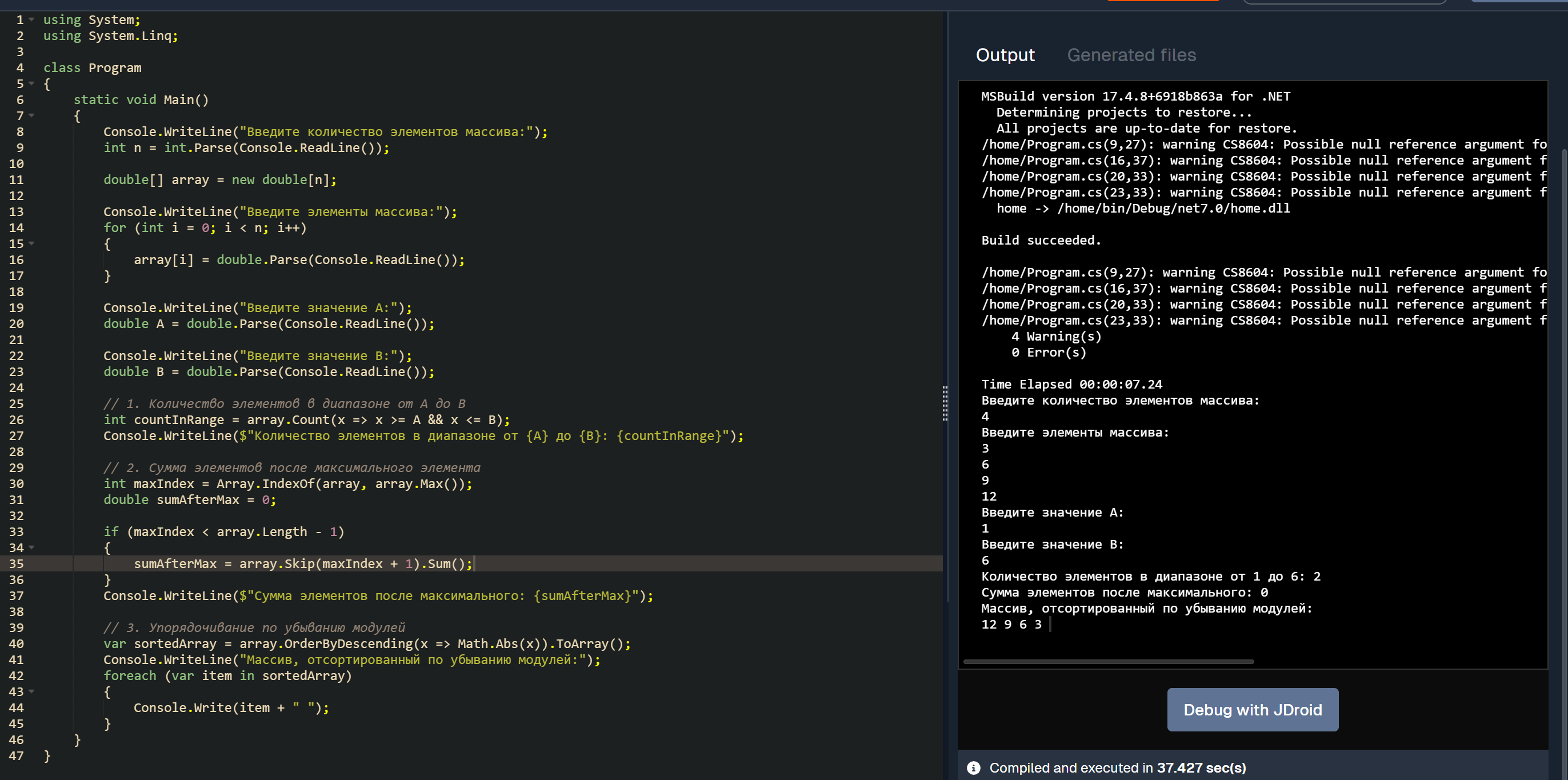
Лабораторная работа 5



Лабораторная работа 6

**using System;**

**class Program**

**{**

**static void Main()**

**{**

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

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

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

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

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

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

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

**{**

**string[] rowValues = Console.ReadLine().Split(' ');**

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

**{**

**matrix[i, j] = int.Parse(rowValues[j]);**

**}**

**}**

**Console.WriteLine("Введите количество элементов для сдвига (n):");**

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

**Console.WriteLine("Выберите направление сдвига (right/down):");**

**string direction = Console.ReadLine().ToLower();**

**if (direction == "right")**

**{**

**ShiftRight(matrix, n);**

**}**

**else if (direction == "down")**

**{**

**ShiftDown(matrix, n);**

**}**

**else**

**{**

**Console.WriteLine("Некорректное направление сдвига!");**

**return;**

**}**

**Console.WriteLine("Результат сдвига:");**

**PrintMatrix(matrix);**

**}**

**// Циклический сдвиг вправо (по строкам)**

**static void ShiftRight(int[,] matrix, int n)**

**{**

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

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

**// Нормализация n (если n > cols, берем остаток от деления)**

**n %= cols;**

**if (n == 0) return; // Сдвиг не нужен**

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

**{**

**int[] tempRow = new int[cols];**

**// Копируем текущую строку во временный массив**

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

**{**

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

**}**

**// Выполняем циклический сдвиг вправо**

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

**{**

**matrix[i, (j + n) % cols] = tempRow[j];**

**}**

**}**

**}**

**// Циклический сдвиг вниз (по столбцам)**

**static void ShiftDown(int[,] matrix, int n)**

**{**

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

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

**// Нормализация n (если n > rows, берем остаток от деления)**

**n %= rows;**

**if (n == 0) return; // Сдвиг не нужен**

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

**{**

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

**// Копируем текущий столбец во временный массив**

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

**{**

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

**}**

**// Выполняем циклический сдвиг вниз**

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

**{**

**matrix[(i + n) % rows, j] = tempCol[i];**

**}**

**}**

**}**

**// Вывод 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] + " ");**

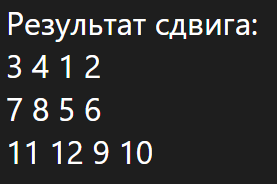
**}**

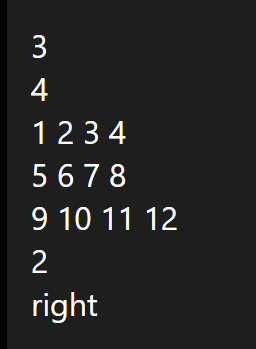
**Console.WriteLine();**

**}**

**}**

**}**





Лабораторная работа 7

**using System;**

**using System.IO;**

**using System.Text;**

**class Program**

**{**

**static void Main()**

**{**

**Console.WriteLine("Выберите способ ввода текста (1 - файл, 2 - консоль):");**

**int choice = int.Parse(Console.ReadLine());**

**string text = "";**

**if (choice == 1)**

**{**

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

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

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

**}**

**else if (choice == 2)**

**{**

**Console.WriteLine("Введите текст (завершите пустой строкой):");**

**string line;**

**while (!string.IsNullOrWhiteSpace(line = Console.ReadLine()))**

**{**

**text += line + Environment.NewLine;**

**}**

**}**

**else**

**{**

**Console.WriteLine("Некорректный выбор");**

**return;**

**}**

**string result = ProcessText(text);**

**Console.WriteLine("Результат:");**

**Console.WriteLine(result);**

**}**

**static string ProcessText(string text)**

**{**

**StringBuilder result = new StringBuilder();**

**char[] vowels = { 'a', 'e', 'i', 'o', 'u' };**

**bool newWord = true;**

**foreach (char c in text)**

**{**

**if (char.IsLetter(c))**

**{**

**if (newWord)**

**{**

**// Проверяем, является ли буква гласной (без учета регистра)**

**bool isVowel = Array.Exists(vowels, v => v == char.ToLower(c));**

**result.Append(isVowel ? char.ToUpper(c) : c);**

**newWord = false;**

**}**

**else**

**{**

**result.Append(c);**

**}**

**}**

**else**

**{**

**result.Append(c);**

**newWord = !char.IsPunctuation(c) && !char.IsDigit(c);**

**}**

**}**

**return result.ToString();**

**}**

**}**

