# **Практическое занятие №4**

*Дата*: 19.04.2023

*Тема:* **Массивы. Алгоритм обработки массивов. Рекурсия**

*Цель:*Научиться использовать массивы, работать с ними и вызывать рекурсию

**Ход работы**

**Весь код**

using System.Data;

namespace лаба\_4

{

public partial class Form1 : Form

{

int[,] matrix = new int[5, 8];

int[] negativeColumns = new int[8];

int negColCount = 0;

public Form1()

{

InitializeComponent();

}

private void tabPage2\_Click\_1(object sender, EventArgs e)

{

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox5.Clear();

textBox6.Clear();

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

int a = int.Parse(textBox1.Text);

int b = int.Parse(textBox2.Text);

int n = int.Parse(textBox3.Text);

int[] arr = new int[n];

arr[0] = a;

arr[1] = b;

for (int i = 2; i < arr.Length; i++)

{

arr[i] = Math.Abs(arr[i - 2] - arr[i - 1]);

Console.WriteLine();

}

string output = "";

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

{

output += $"{i + 1} элемент массива = {arr[i]}\r\n";

}

textBox4.Text = output;

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

// Проверка корректности введенного размера массива

if (!int.TryParse(textBox5.Text, out int n) || n <= 0)

{

textBox6.Text = "Ошибка ввода: размер массива должен быть положительным целым числом.";

return;

}

// Создание и заполнение массива

int[] arr = new int[n];

Random rnd = new Random();

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

{

int rndNum = rnd.Next(1, 1000);

// Проверка уникальности числа

while (Array.IndexOf(arr, rndNum) != -1)

{

rndNum = rnd.Next(1, 1000);

}

arr[i] = rndNum;

}

// Поиск первого элемента, ломающего возрастающую последовательность

int idx = -1;

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

{

if (arr[i] >= arr[i + 1])

{

idx = i + 2;

break;

}

}

// Вывод результата в соответствующий TextBox

if (idx == -1)

{

textBox6.Text = "Элементы удачно создали возрастающую последовательность\r\n";

}

else

{

textBox6.Text = $" {idx.ToString()} элемент массива разрушил возрастающую последовательность\r\n";

}

string output = "";

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

{

output += $"{i + 1} элемент массива = {arr[i]} \r\n";

}

textBox6.Text += output;

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void button3\_Click(object sender, EventArgs e)

{

this.Width = 942;

button4.Enabled = true;

textBox7.Clear();

Random random = new Random();

// Заполняем матрицу случайными числами

for (int row = 0; row < 5; row++)

{

for (int col = 0; col < 8; col++)

{

matrix[row, col] = random.Next(-10, 11);

}

}

// Установка названия таблицы

dataGridView1.DataSource = null;

dataGridView1.Rows.Clear();

dataGridView1.Columns.Clear();

dataGridView1.Refresh();

dataGridView1.DataSource = GetDataTable(matrix, "matrix");

}

// Метод, преобразующий массив в DataTable

private DataTable GetDataTable(int[,] matrix, string tableName)

{

DataTable dt = new DataTable(tableName);

// Добавление столбцов

for (int i = 0; i < matrix.GetLength(1); i++)

{

dt.Columns.Add(new DataColumn($"Cтолбец {i}", typeof(int)));

}

// Добавление строк

for (int i = 0; i < matrix.GetLength(0) ; i++)

{

DataRow dr = dt.NewRow();

for (int j = 0; j < matrix.GetLength(1); j++)

{

dr[j] = matrix[i, j];

}

dt.Rows.Add(dr);

}

return dt;

}

private void button4\_Click(object sender, EventArgs e)

{

button4.Enabled = false;

textBox7.Text = "Матрица имеет отрицательные элементы в столбце: ";

for (int col = 0; col < 8; col++)

{

bool hasNegative = false;

for (int row = 0; row < 5; row++)

{

if (matrix[row, col] < 0)

{

hasNegative = true;

break;

}

}

if (hasNegative)

{

textBox7.Text += col.ToString() + " ";

}

}

}

private void tabPage1\_Click(object sender, EventArgs e)

{

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox5.Clear();

textBox6.Clear();

}

private void Form1\_Load(object sender, EventArgs e)

{

tabPage1.Text = "Задание 1";

tabPage2.Text = "Задание 3";

tabPage3.Text = "Задание 2";

this.Text = "Лабораторная работа #4";

}

private void TabControl(object sender, EventArgs e)

{

foreach (TabPage page in tabControl1.TabPages)

{

foreach (Control control in page.Controls)

{

if (control is TextBox textBox)

{

textBox.Clear();

this.Width = 656;

dataGridView1.Columns.Clear();

}

}

}

}

private void textBox6\_TextChanged(object sender, EventArgs e)

{

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

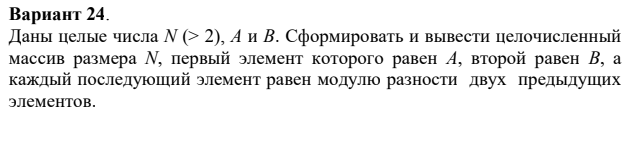
}

}

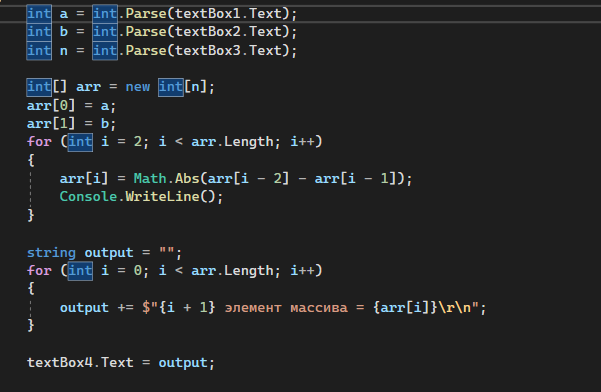
}

**Задание 1**

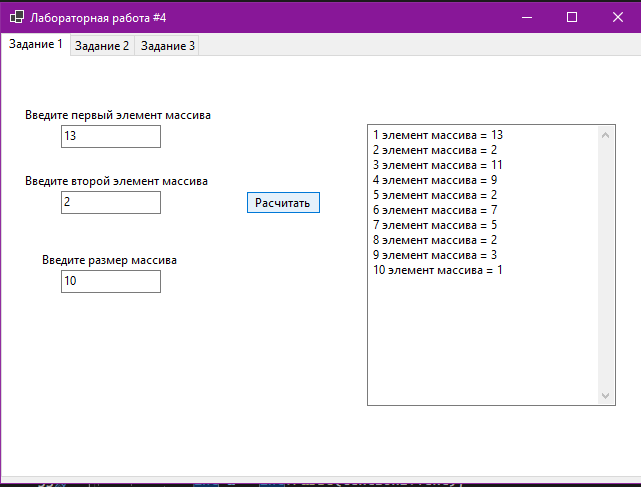
**Условие**

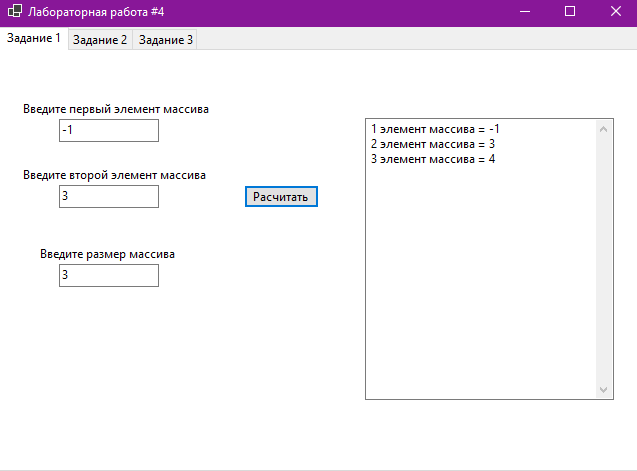


**Исходный код**



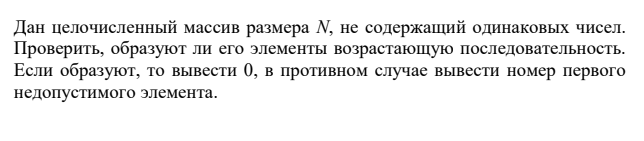
**Результат**



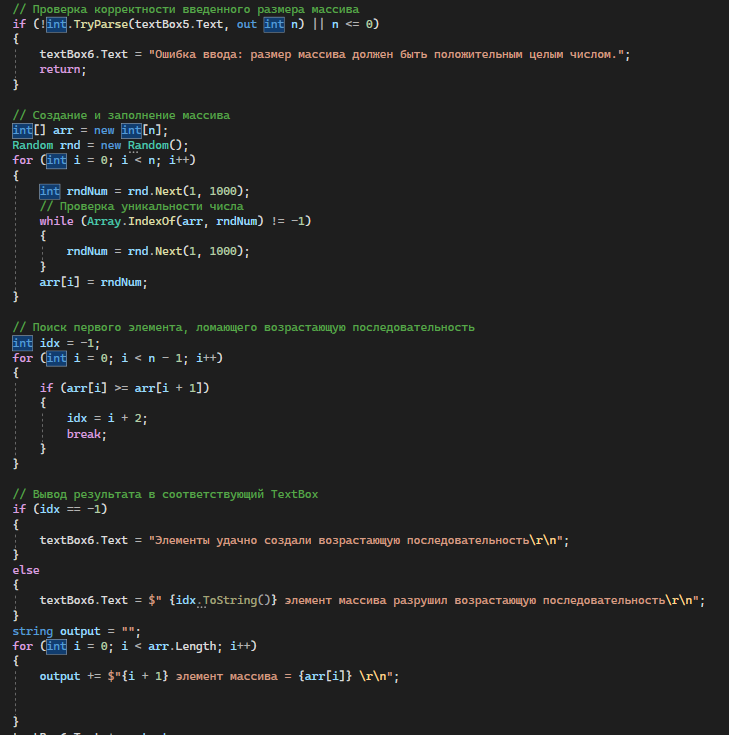


**Задание 2**

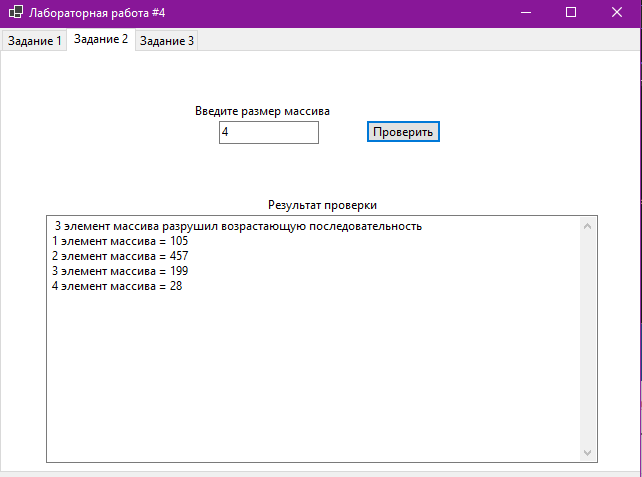
**Условие**

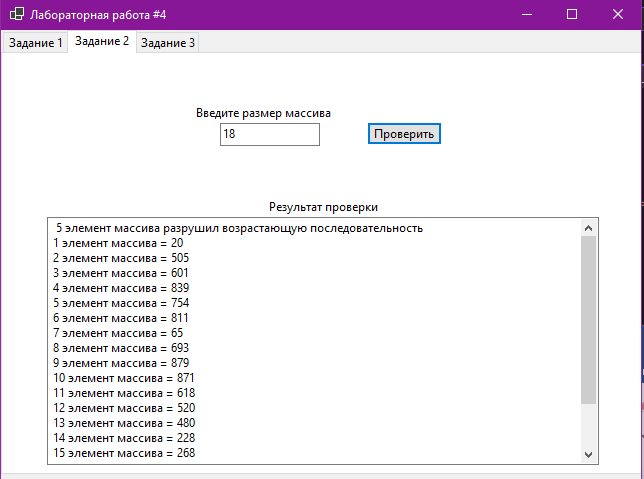


**Исходный код**



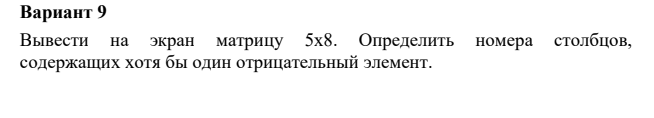
**Результат**



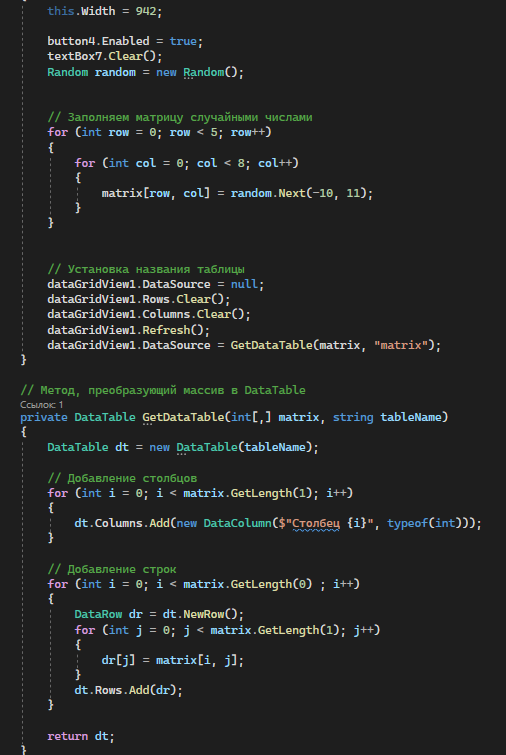


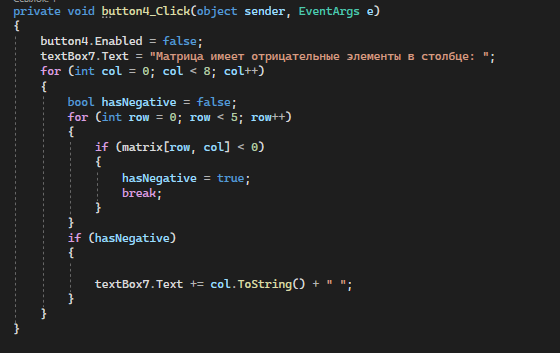
**Задание 3**

**Условие**



**Исходный код**





**Результат**

