Лабораторная работа №5

«Работа с двумерными массивами»

**Задание**

Дан двумерный массив положительных и отрицательных чисел. Разработать метод, позволяющий найти и вывести количество элементов исходной матрицы, лежащих выше главной диагонали, в которых хотя бы одно число делится на 3. Разработать метод, который формирует новый массив, из тех элементов исходной матрицы, значения которых больше найденного числа.

Требуется создать приложение, запрашивающее у пользователя через **InputBox** количество строк и столбцов массива, а также минимально и максимальное возможное значения, которыми он будет заполнен. Далее необходимо посчитать количество чисел которые делятся на 3 и находятся выше главной диагонали. После этого создать новый одномерный массив который будет состоять только из чисел которые больше вычисленного ранее количества чисел подходящих под условие.

Так же необходимо разработать кнопки для сохранения полученных данных в Word, Excel и текстовый файл.

Для выполнения данного задания необходимо разработать следующие методы:

Методы визуальные, которые сделаны для удобства пользователя самостоятельно просматривать числа подходящие под условие:

public static void SelectMainDiagonal(DataGridView Table, int length) – Метод которые выделяет главную диагональ для удобства пользователя. Принимает таблицу DataGridView и длину передаваемой таблицы.

public static void Sem2\_Lab5\_Select\_Replaced\_Nums(int[,] main, int num, DataGridView Table) – Метод который выделяет числа подходящие под условие задачи. Принимает массив с случайно сгенерированными числами но ограниченным минимально и максимально допустимыми значениями пользователем, число для проверки условия и таблицу для выделения значений.

public static void Sem2\_Lab5\_Off\_selected\_cells(int length, DataGridView TableMain, int length2, DataGridView TableTwink) – Метод убирает выделение которое пользователь мог использовать ранее. Принимает длину основной, вспомогательной таблицы и сами таблицы.

Методы вычислений:

public static void Sem2\_Lab5\_Fill\_Array\_RND\_Num(int[,] mas, int min, int max) – Метод для заполнения двумерного массива случайными числами с ограниченными минимальными и максимальными значениями которые указал пользователь. Принимает двумерный массив, минимальное и максимальное значение которое вводить пользователь.

public static void Sem2\_Lab5\_Fill\_corner\_DGV(int length, DataGridView Table) – Метод для заполнения границ таблицы. Принимает длину таблицы и таблицу.

public static void Sem2\_Lab5\_Enter\_Mas\_in\_DGV(int[,] mas, DataGridView Table) – Метод для вывода массива в таблицу. Принимает двумерный массив и таблицу для вывода массива.

public static void Sem2\_Lab5\_Enter\_Mas\_in\_DGV(int[] mas, DataGridView Table) – Перегруженный метод для вывода массива в таблицу, который принимает одномерный массив и таблицу для вывода результирующего массива.

public static int[] Sem2\_Lab5\_Count\_Condition(int[,] main) – Метод для подсчета количества чисел которые делятся на 3 и находятся выше главной диагонали, и высчитывания количества чисел которые больше посчитанного ранее количества чисел. Принимает двумерный массив.

public static int[] Sem2\_Lab5\_Usually(int arr\_size, int[,] main, int compare) – Метод который возвращает одномерный массив с числами которые подходят под условие задачи. Принимает количество чисел, которые подходят под условие для генерации нужного размера одномерного массива, двумерный массив и число выше которого должны быть числа по условию задачи.

public static int Get\_Length(int[,] mas) – Вспомогательный метод для упрощения кода. Возвращает длину двумерного массива путем вычисления квадратного корня. Принимает двумерный массив, возвращает длину массива.

public static void Sem2\_Lab5\_Save\_TXT(int[,] array, int[] array2) – Метод для сохранения вычисленных данных в текстовый файл. Принимает двумерный массив и двумерный массив.

public static void Sem2\_Lab5\_Save\_Word(int[,] main, int[] twin) Метод для сохранения вычисленных данных в Word файл. Принимает двумерный массив и двумерный массив.

public static void Sem2\_Lab5\_Save\_Excel(int[,] main, int[] twin) Метод для сохранения вычисленных данных в Excel файл. Принимает двумерный массив и двумерный массив.

**1.1 Разработка алгоритма решения задачи.**

Схема алгоритма событийной процедуры решения задачи представлена на рисунке 1.

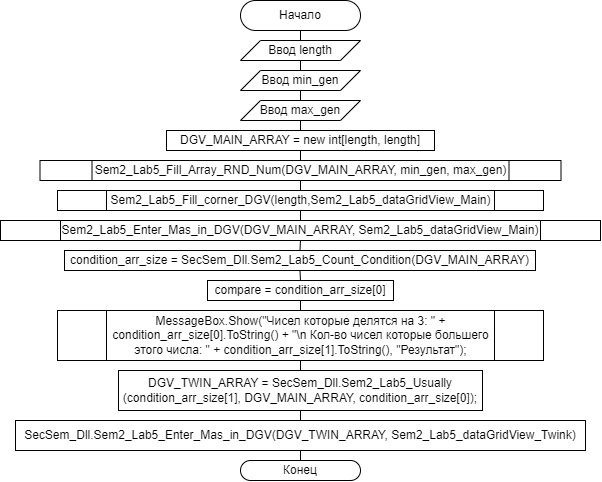


Рисунок 1 – Схема алгоритма процедуры.

Схема алгоритма, выполняющего заполнение массива целыми числами в определённом диапазоне, представлена на рисунке 2.

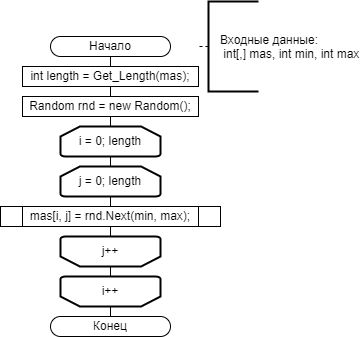


Рисунок 2 – Схема алгоритма заполнения массива.

Схема алгоритма, выполняющего вывод двумерного массива в DataGridView, представлена на рисунке 3.

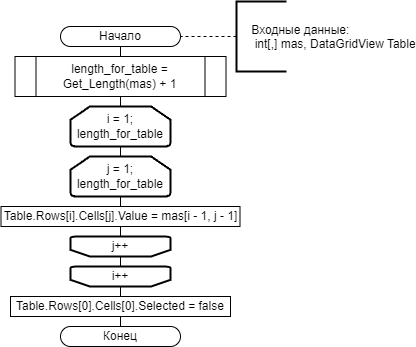


Рисунок 3 – Вывод массива в таблицу.

Схема алгоритма который считает количество чисел подходящих под условие представлена на рисунке 4.

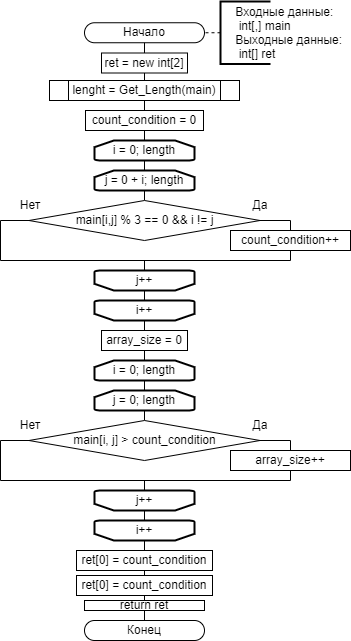


Рисунок 4 – Схема алгоритма высчитывающего кол-во чисел.

Схема алгоритма который нумерует края таблицы представлена на рисунке 5.

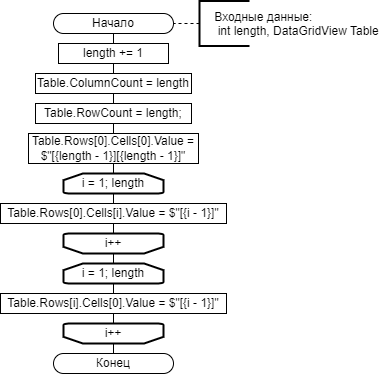


Рисунок 5 – Схема алгоритма нумерации краев таблицы.

Схема вспомогательного алгоритма, который возвращает длину двумерного массива представлена на рисунке 6.

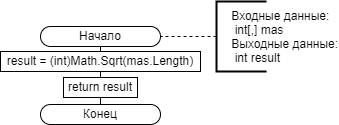


Рисунок 6 – Схема алгоритма вычисления длины двумерного массива.

Схема алгоритма, который выводит одномерный массив в DataGridView представлена на рисунке 7.

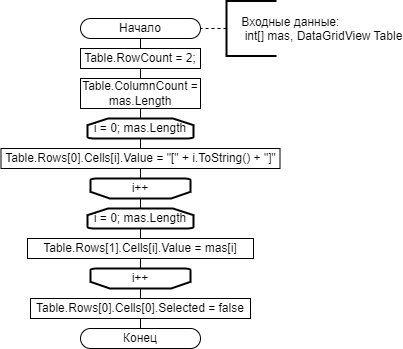


Рисунок 7 – Схема алгоритма вывода массива.

Схема алгоритма, который очищает таблицы от выбора пользователя или использования функций для выделения представлена на рисунке 8.



Рисунок 8 – Схема алгоритма очистки таблиц.

Схема алгоритма, который выделяет числа подходящие под условие задачи представлена на рисунке 9.

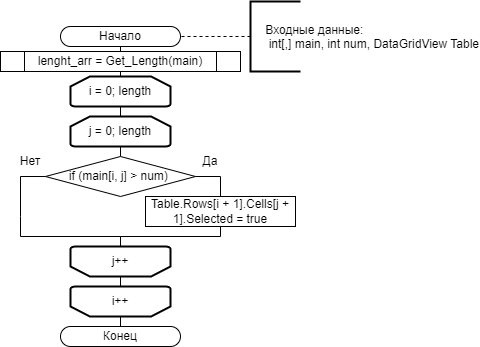


Рисунок 9 – Схема алгоритма выделения подходящих чисел.

Схема алгоритма выделения главной диагонали для пользователя представлена на рисунке 10.

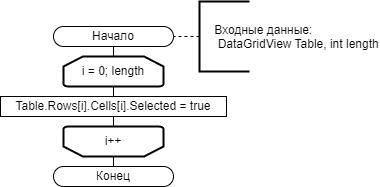


Рисунок 10 – Схема алгоритма выделения главной диагонали.

Схема алгоритма заполняющего вспомогательный массив по условию задачи представлена на рисунке 11.

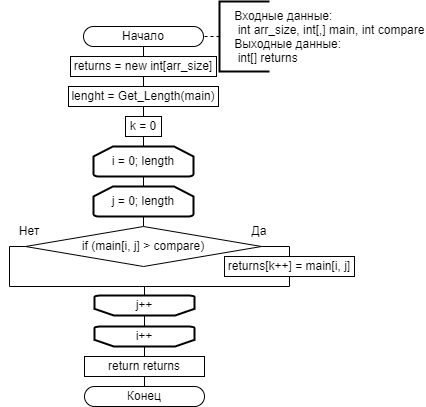


Рисунок 11 – Схема алгоритма заполнения массива.

**1.2 Разработка программного кода.**

Код находящийся в DLL-библиотеке:

using System;

using System.Windows.Forms;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Diagnostics;

using System.IO;

using Excel = Microsoft.Office.Interop.Excel;

using Word = Microsoft.Office.Interop.Word;

using Microsoft.Office.Interop.Excel;

using System.Security.Cryptography.X509Certificates;

using Microsoft.Office.Interop.Word;

namespace Second\_Semestr\_Dll

{

public class SecSem\_Dll

{

public static void Enter\_massiv(int[] mas, int length, int min, int max)

{

// Убрать length так как легко крашнуть

Random rnd = new Random();

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

mas[i] = rnd.Next(min, max);

}

public static void Output\_massiv(int[] mas, int length, DataGridView Table)

{

Table.ColumnCount = length;

Table.RowCount = 2;

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

{

Table.Rows[0].Cells[i].Value = "[" + i + "]";

Table.Rows[1].Cells[i].Value = mas[i];

}

}

public static void Sem2\_Lab3\_Calc\_Geometric(int[] mas, out double result)

{

double geometric = 1.0;

double count = 0;

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

{

if (mas[i] % 2 == 0)

{

geometric \*= mas[i];

count++;

}

}

// (x1\*x2\*x3\*x4\*x5) ^ (1/5);

result = Math.Pow(geometric, (1 / count));

}

public static void Sem2\_Lab3\_Count\_Condition(int[] mas, double result, out int index)

{

// Вывести четные числа которые больше геометрического

index = 0;

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

if (mas[i] > result)

index++;

}

public static void Sem2\_Lab3\_Calculate(int[] mas, double result, int index, out int[] output)

{

output = new int[index];

int k = 0;

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

if (mas[i] > result)

output[k++] = mas[i];

}

public static void Sem2\_Lab3\_Save\_Txt(int[] array, int[] array2)

{

StreamWriter streamWriter = File.CreateText("Массивы.txt");

streamWriter.WriteLine("Исходный массив:");

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

{

streamWriter.WriteLine(array[i].ToString());

}

streamWriter.WriteLine("Конечный массив:");

for (int i = 0; i < array2.Length; i++)

{

streamWriter.WriteLine(array2[i].ToString());

}

streamWriter.Close();

Process.Start("Массивы.txt");

}

public static void Sem2\_Lab3\_Save\_Excel(int[] array, int[] array2)

{

Excel.Application excel = new Excel.Application();

Excel.Workbook WB = excel.Workbooks.Add();

Excel.Worksheet ws = WB.Worksheets[1];

ws.Name = "Исходный массив";

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

{

ws.Cells[2, i + 1] = "[" + i + "]";

ws.Cells[3, i + 1] = array[i].ToString();

}

Excel.Worksheet ws2 = WB.Worksheets.Add();

ws2.Name = "Конечный массив";

for (int i = 0; i < array2.Length; i++)

{

ws2.Cells[2, i + 1] = "[" + i + "]";

ws2.Cells[3, i + 1] = array2[i].ToString();

}

excel.Visible = true;

excel.UserControl = true;

}

public static void Sem2\_Lab3\_Save\_Word(int[] array, int[] array2)

{

Word.Application word = new Word.Application();

Word.Document doc = word.Documents.Add();

Word.Range range = doc.Range();

range.Text = "Исходный массив:";

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

{

range.Text += "[" + i + "] " + array[i].ToString();

}

range.Text += "\nКонечный массив:";

for (int i = 0; i < array2.Length; i++)

{

range.Text += "[" + i + "] " + array2[i].ToString() + "";

}

word.Visible = true;

}

public static void SelectMainDiagonal(DataGridView Table, int length)

{

for (int i = 0; i < length + 1; i++)

Table.Rows[i].Cells[i].Selected = true;

}

public static void Sem2\_Lab5\_Select\_Replaced\_Nums(int[,] main, int num, DataGridView Table)

{

int lenght\_arr = Get\_Length(main);

for (int i = 0; i < lenght\_arr; i++)

for (int j = 0; j < lenght\_arr; j++)

if (main[i, j] > num)

Table.Rows[i + 1].Cells[j + 1].Selected = true;

}

public static void Sem2\_Lab5\_Off\_selected\_cells(int length, DataGridView TableMain, int length2, DataGridView TableTwink)

{

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

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

TableMain.Rows[i].Cells[j].Selected = false;

for (int i = 0; i < 2; i++)

for (int j = 0; j < length2; j++)

TableTwink.Rows[i].Cells[j].Selected = false;

}

public static void Sem2\_Lab5\_Fill\_Array\_RND\_Num(int[,] mas, int min, int max)

{

int length = Get\_Length(mas);

Random rnd = new Random();

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

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

mas[i, j] = rnd.Next(min, max);

}

public static void Sem2\_Lab5\_Fill\_corner\_DGV(int length, DataGridView Table)

{

length += 1;

Table.ColumnCount = length;

Table.RowCount = length;

Table.Rows[0].Cells[0].Value = $"[{length - 1}][{length - 1}]";

for (int i = 1; i < length; i++)

Table.Rows[0].Cells[i].Value = $"[{i - 1}]";

for (int i = 1; i < length; i++)

Table.Rows[i].Cells[0].Value = $"[{i - 1}]";

}

public static void Sem2\_Lab5\_Enter\_Mas\_in\_DGV(int[,] mas, DataGridView Table)

{

int length\_for\_table = Get\_Length(mas) + 1;

for (int i = 1; i < length\_for\_table; i++)

for (int j = 1; j < length\_for\_table; j++)

Table.Rows[i].Cells[j].Value = mas[i - 1, j - 1];

Table.Rows[0].Cells[0].Selected = false;

}

public static void Sem2\_Lab5\_Enter\_Mas\_in\_DGV(int[] mas, DataGridView Table)

{

Table.RowCount = 2;

Table.ColumnCount = mas.Length;

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

Table.Rows[0].Cells[i].Value = "[" + i.ToString() + "]";

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

Table.Rows[1].Cells[i].Value = mas[i];

Table.Rows[0].Cells[0].Selected = false;

}

public static int[] Sem2\_Lab5\_Count\_Condition(int[,] main)

{

int[] ret = new int[2];

int lenght = Get\_Length(main);

int count\_condition = 0;

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

for (int j = 0 + i;j < lenght; j++)

if (main[i,j] % 3 == 0 && i != j)

count\_condition++;

int array\_size = 0;

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

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

if (main[i, j] > count\_condition)

array\_size++;

// Чисел которые делятся на 3

ret[0] = count\_condition;

ret[1] = array\_size;

return ret;

}

public static int[] Sem2\_Lab5\_Usually(int arr\_size, int[,] main, int compare)

{

int[] returns = new int[arr\_size];

int lenght = Get\_Length(main);

int k = 0;

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

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

if (main[i, j] > compare)

returns[k++] = main[i, j];

return returns;

}

public static int Get\_Length(int[,] mas)

{

int result = (int)Math.Sqrt(mas.Length);

return result;

}

public static void Sem2\_Lab5\_Save\_TXT(int[,] array, int[] array2)

{

int length = (int)Math.Sqrt(array.Length);

StreamWriter streamWriter = File.CreateText("Массивы.txt");

streamWriter.WriteLine("Исходный массив:");

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

{

string strAdd = "";

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

strAdd += "[" + array[i, j].ToString() + "]" + ' ';

streamWriter.WriteLine(strAdd);

}

streamWriter.WriteLine("Конечный массив:");

string strAdd2 = "";

for (int i = 0; i < array2.Length; i++)

{

strAdd2 += "[" + array2[i].ToString() + "]" + ' ';

}

streamWriter.WriteLine(strAdd2);

streamWriter.Close();

Process.Start("Массивы.txt");

}

public static void Sem2\_Lab5\_Save\_Word(int[,] main, int[] twin)

{

Word.Application word = new Word.Application();

Word.Document doc = word.Documents.Add();

Word.Range range = doc.Range();

Word.Table table = doc.Tables.Add(range, main.GetLength(0), main.GetLength(1));

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

{

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

{

table.Cell(i + 1, j + 1).Range.Text = main[i, j].ToString();

}

}

word.Visible = true;

}

public static void Sem2\_Lab5\_Save\_Excel(int[,] main, int[] twin)

{

int lenght = Get\_Length(main);

Microsoft.Office.Interop.Excel.Application excelApp = new Microsoft.Office.Interop.Excel.Application();

Microsoft.Office.Interop.Excel.Workbook workBook;

Microsoft.Office.Interop.Excel.Worksheet workSheet1, workSheet2;

workBook = excelApp.Workbooks.Add();

workSheet1 = (Microsoft.Office.Interop.Excel.Worksheet)workBook.Worksheets.get\_Item(1);

workSheet1.Name = "Исходный массив";

excelApp.Cells[1, 1] = "Исходный массив";

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

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

{

excelApp.Cells[i + 3, 2] = $"[{i}]";

excelApp.Cells[2, j + 3] = $"[{j}]";

excelApp.Cells[i + 3, j + 3] = main[i, j];

}

workSheet2 = workBook.Worksheets.Add();

int k = 0;

workSheet2.Name = "Результирующий массив";

excelApp.Cells[1, 1] = "Результирующий массив";

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

{

workSheet2.Cells[3, i+2] = "[" + (i+1) + "]";

workSheet2.Cells[4, i+2] = twin[i];

}

excelApp.Visible = true;

excelApp.UserControl = true;

}

}

}

Основной код программы:  
using Microsoft.Office.Interop.Word;

using Microsoft.VisualBasic;

using Second\_Semestr\_Dll;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Лабораторная\_работа

{

public partial class Sem2\_CalcLab5 : Form

{

public Sem2\_CalcLab5()

{

InitializeComponent();

}

private void Sem2\_Lab5\_exit\_Click(object sender, EventArgs e)

{

this.Close();

}

int[,] DGV\_MAIN\_ARRAY;

int[] DGV\_TWIN\_ARRAY;

int length\_select = 0;

int compare = 0;

private void Sem2\_Lab5\_calc\_button\_Click(object sender, EventArgs e)

{

if (length\_select != 0)

SecSem\_Dll.Sem2\_Lab5\_Off\_selected\_cells(length\_select,Sem2\_Lab5\_dataGridView\_Main, DGV\_TWIN\_ARRAY.Length, Sem2\_Lab5\_dataGridView\_Twink);

string length\_str = Interaction.InputBox(Prompt: "Введите количество элементов массива для генерации", Title: "Размерность массива", DefaultResponse: "6");

// Если нажали кнопку отмена

if (length\_str == "") { return; }

string min\_gen\_str = Interaction.InputBox(Prompt: "Нижняя граница генерации чисел", Title: "Минимальное значение", DefaultResponse: "-10");

if (min\_gen\_str == "") { return; }

string max\_gen\_str = Interaction.InputBox(Prompt: "Верхняя граница генерации чисел", Title: "Максимально значение", DefaultResponse: "20");

if (max\_gen\_str == "") { return; }

int length = Convert.ToInt32(length\_str);

length\_select = length;

int min\_gen = Convert.ToInt32(min\_gen\_str);

int max\_gen = Convert.ToInt32(max\_gen\_str);

if (max\_gen < min\_gen) { MessageBox.Show("Верхняя граница генерации чисел должна быть больше нижней. Программа остановлена"); return; }

// Если ввели максимальное число меньше чем меньшее

DGV\_MAIN\_ARRAY = new int[length, length]; // Определяем двойной массив

// Заполняем массив

SecSem\_Dll.Sem2\_Lab5\_Fill\_Array\_RND\_Num(DGV\_MAIN\_ARRAY, min\_gen, max\_gen);

// Рисуем края mianDGV

SecSem\_Dll.Sem2\_Lab5\_Fill\_corner\_DGV(length, Sem2\_Lab5\_dataGridView\_Main);

// Выводим массив на mainDGV

SecSem\_Dll.Sem2\_Lab5\_Enter\_Mas\_in\_DGV(DGV\_MAIN\_ARRAY, Sem2\_Lab5\_dataGridView\_Main);

//Считаем по условию mainDGV

int[] condition\_arr\_size = SecSem\_Dll.Sem2\_Lab5\_Count\_Condition(DGV\_MAIN\_ARRAY);

compare = condition\_arr\_size[0];

// Выводим результат MsgBox

MessageBox.Show("Чисел которые делятся на 3: " + condition\_arr\_size[0].ToString() + "\n Кол-во чисел которые большего этого числа: " + condition\_arr\_size[1].ToString(), "Результат");

DGV\_TWIN\_ARRAY = SecSem\_Dll.Sem2\_Lab5\_Usually(condition\_arr\_size[1], DGV\_MAIN\_ARRAY, condition\_arr\_size[0]);

SecSem\_Dll.Sem2\_Lab5\_Enter\_Mas\_in\_DGV(DGV\_TWIN\_ARRAY, Sem2\_Lab5\_dataGridView\_Twink);

}

private void Sem2\_Lab5\_Select\_Main\_Diagonal\_dgv\_main\_Click(object sender, EventArgs e)

{

SecSem\_Dll.SelectMainDiagonal(Sem2\_Lab5\_dataGridView\_Main, length\_select);

}

private void Sem2\_Lab5\_Select\_Nums\_Click(object sender, EventArgs e)

{

SecSem\_Dll.Sem2\_Lab5\_Select\_Replaced\_Nums(DGV\_MAIN\_ARRAY, compare,Sem2\_Lab5\_dataGridView\_Main);

}

private void Sem2\_Lab5\_Off\_selected\_cells\_Click(object sender, EventArgs e)

{

SecSem\_Dll.Sem2\_Lab5\_Off\_selected\_cells(length\_select+1, Sem2\_Lab5\_dataGridView\_Main, DGV\_TWIN\_ARRAY.Length, Sem2\_Lab5\_dataGridView\_Twink);

}

private void Sem2\_Lab5\_Save\_TXT\_button\_Click(object sender, EventArgs e)

{

SecSem\_Dll.Sem2\_Lab5\_Save\_TXT(DGV\_MAIN\_ARRAY, DGV\_TWIN\_ARRAY);

}

private void Sem2\_Lab5\_Save\_Word\_button\_Click(object sender, EventArgs e)

{

SecSem\_Dll.Sem2\_Lab5\_Save\_Word(DGV\_MAIN\_ARRAY, DGV\_TWIN\_ARRAY);

}

private void Sem2\_Lab5\_Save\_Excel\_button\_Click(object sender, EventArgs e)

{

SecSem\_Dll.Sem2\_Lab5\_Save\_Excel(DGV\_MAIN\_ARRAY, DGV\_TWIN\_ARRAY);

}

}

}

**1.3 Демонстрация работы программы.**

На рисунке 12 представлены условия задачи.

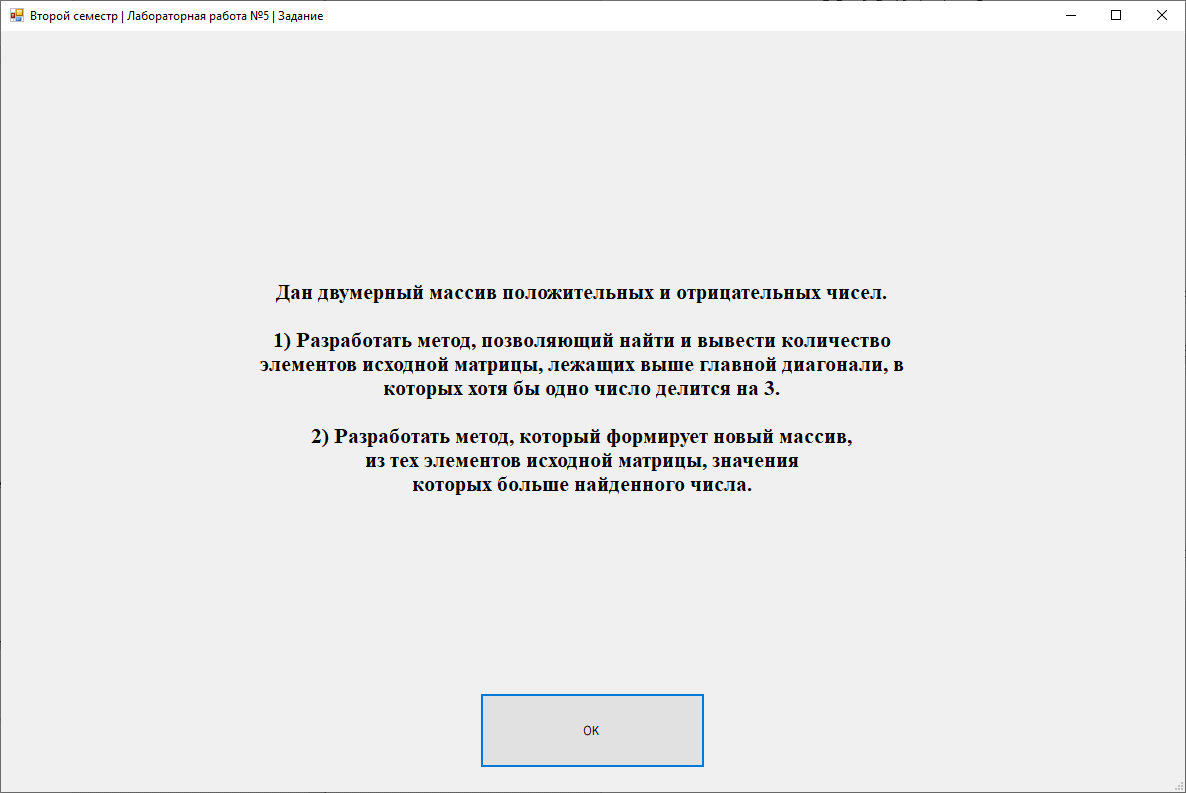


Рисунок 12 – Условия задачи.

На рисунке 13 демонстрация работы программы.

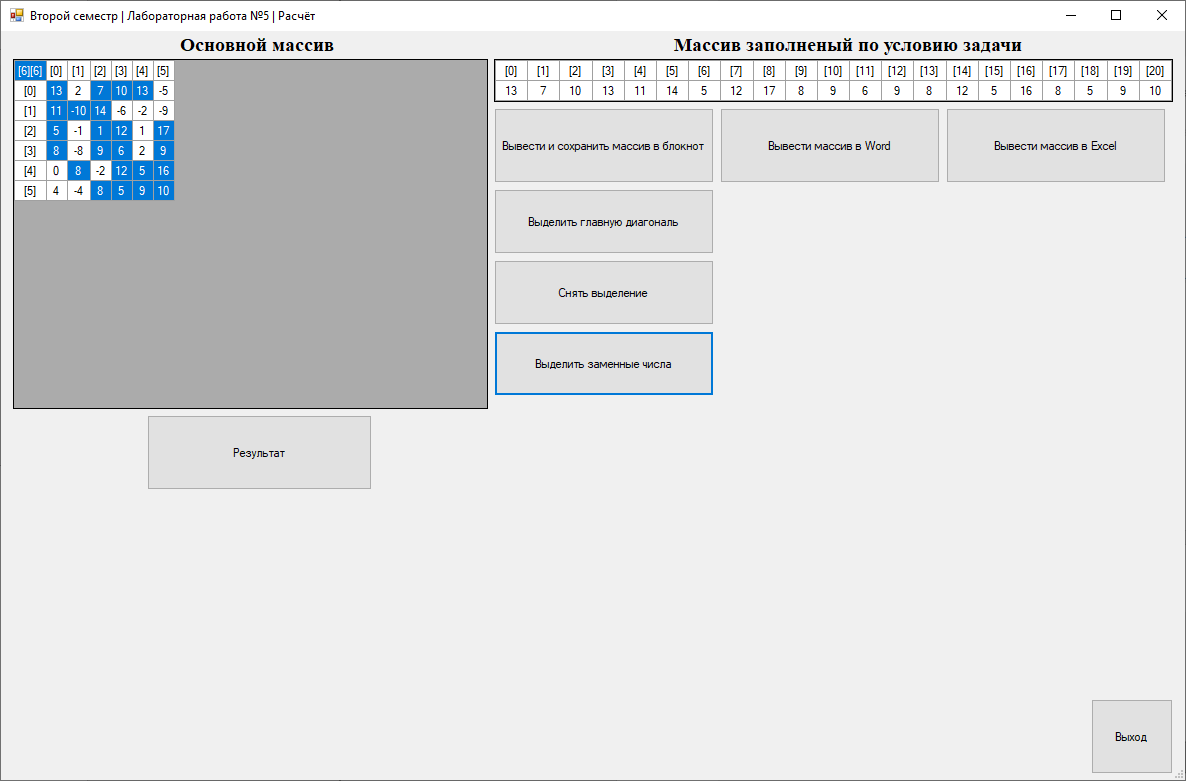


Рисунок 13 – Демонстрация работы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гуриков, С. Р. Введение в программирование на языке Visual C#: учебное пособие / С. Р. Гуриков. — Москва: ФОРУМ: ИНФРА-М, 2019. — 447 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-540-0. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1012397> (дата обращения: 01.06.2024).

2. ГОСТ 7.32-2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления" (введен в действие Приказом Росстандарта от 24.10.2017 N 1494-ст) (дата обращений 01.06.2024).

3. ГОСТ 19.701-90 (ИСО 5807-85) “Единая система программной документации СХЕМЫ АЛГОРИТМОВ, ПРОГРАММ, ДАННЫХ И СИСТЕМ Обозначения условные и правила выполнения”. (дата обращения 01.06.2024).