**Лабораторная работа № 6**

**«ЗАПИСЬ ДВУМЕРНЫХ МАССИВОВ ВО ВНЕШНИЕ ПРИЛОЖЕНИЯ»**

1. **Постановка задачи**

Дан двумерный массив положительных и отрицательных чисел. Разработать метод, позволяющий найти и вывести наименьшую сумму нечетных элементов на двух диагоналях: главной и побочной. Разработать метод, который формирует новый массив, из тех элементов исходной матрицы, значения которых больше найденного числа.

Кроме того, следует организовать запись одномерного массива в редактор MICROSOFT WORD, который получается в результате выполнения программы по двумерным массивам. Также нужно выполнить задание по алгоритмам обработки двумерных массивов: транспонирование матрицы, вставка строки из значения, обмен столбцов матрицы, удаление заданного столбца матрицы, нахождение количества положительных элементов и нахождение строки с минимальным элементом.

Для этого было необходимо разработать следующие методы:

* Метод ***Vvod()***, предназначенный для ввода данных;
* Метод ***Vivod()***, предназначенный для вывода данных;
* Метод ***ArrayGenerate()***, предназначенный для создания массива, состоящего из случайного набора положительных и отрицательных чисел;
* Метод ***Output\_mas()***, предназначенный для вывода массива в таблицу;
* Метод ***Out\_odnomas()***, предназначенный для вывода одномерного массива в таблицу;
* Метод ***summ(),*** предназначенный для поиска наименьшей суммы нечетных чисел на главной и побочной диагоналях;
* Метод ***Generate()***, формирующий новый одномерный массив, из тех элементов исходного массива, значения которых больше наименьшей нечетной суммы главной и побочной диагоналей;
* Метод ***Kol()***, рассчитывает количество значений которые больше наименьшей нечетной суммы главной и побочной диагоналей;
* Метод ***ZapisWordIsx()***, предназначенный для записи одномерного массива в Microsoft Word;
* Метод ***Zad4()***, предназначенный для транспонирования двумерного массива;
* Метод ***Zad6()***, предназначенный для вставки строки из значения;
* Метод ***Zad8()***, предназначенный для обмена столбцов матрицы;
* Метод ***Zad11()***, предназначенный для удаления столбца матрицы;
* Метод ***Zad13()***, предназначенный для нахождения количества положительных элементов;
* Метод ***Zad15()***, предназначенный для нахождения номера строки с минимальным элементом;
* Метод ***add*()**, предназначенный для создания базы данных;
* Метод ***CreateStructBD*()**, предназначенный для создания структуры базы данных;
* Метод ***CommBD*()**, предназначенный для возвращения числа нужных строк;
* Метод ***AddToBD*()**, предназначенный для записи двумерных массивов в базу данных;
* Метод***ZapisBloknot*()**, предназначенный для записи двумерных массивов в Блокнот;
* Метод ***Zap\_Excel*()**, предназначенный для записи двумерных массивов в Excel;
* Метод ***ZapisWord*()**, предназначенный для записи двумерных массивов в Word.

1. **Разработка проекта**
2. Разработка графического интерфейса пользователя

Форма интерфейса пользователя представлена на рисунке 1.

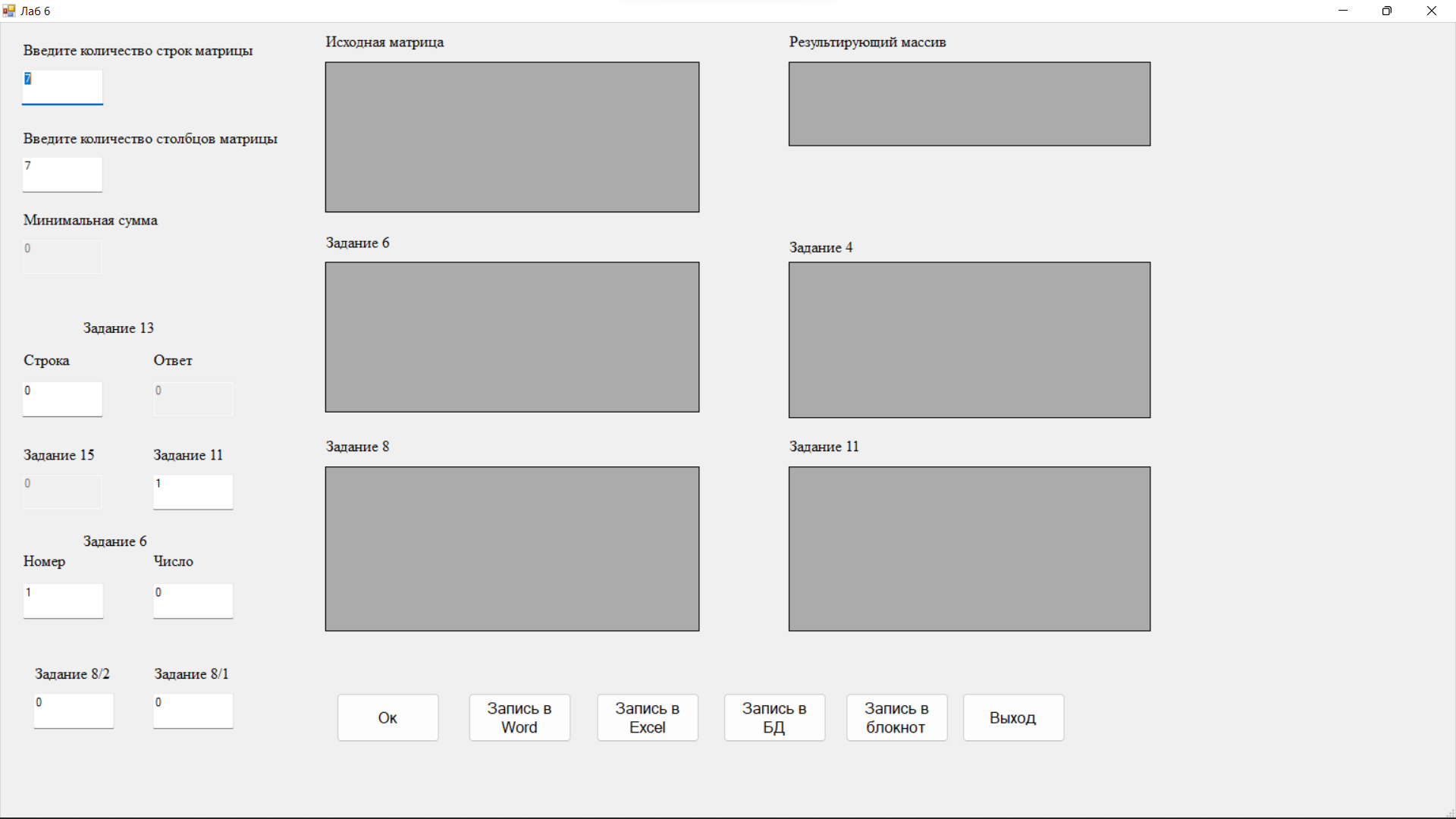


Рисунок 1 - Графический интерфейс пользователя

* 1. Написание программного кода

В основной части проекта подпрограммы-методы вызываются из класса ***class2*** библиотеки ***3sem.***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AnP\_proj

{

public partial class Form13 : Form

{

public Form13()

{

InitializeComponent();

}

public static int[,] arr;

public static int[] rezmas;

public static int stroki;

public static int stolbi;

public static int length;

public int min;

public int nomer;

public int chislo;

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

{

Application.Exit();

}

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

{

\_3sem.Class2.ZapisWordIsx(stroki, stolbi, arr);

}

private void button5\_Click(object sender, EventArgs e)

{

\_3sem.Class2.add();

\_3sem.Class2.CreateStructBD(stolbi);

string commandString = "Provider = Microsoft.Jet.OLEDB.4.0; Data Source = C:\\Users\\Bulat\\Desktop\\Мои проекты\\Курсовые\\АиП\\\_Курсовая\_db.accdb";

var conn = new OleDbConnection(commandString);

\_3sem.Class2.CommBD(commandString, conn);

\_3sem.Class2.AddToBD(stroki, arr, stolbi);

}

private void button6\_Click(object sender, EventArgs e)

{

\_3sem.Class2.ZapisBloknot(arr, stroki, stolbi);

}

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

{

\_3sem.Class2.Zap\_Excel(arr, rezmas, stroki, stolbi);

}

}

}

Все подпрограммы-методы размещаются в классе (***class2***).

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

using System.Globalization;

using Microsoft.VisualBasic;

using System.Diagnostics;

using iTextSharp.text;

using iTextSharp.text.pdf;

using System.Data.OleDb;

using Microsoft.Office;

namespace \_3sem

{

public class Class2

{

public static void ZapisWordIsx(int stroki,int stolbi, int[,] mas)

{

Microsoft.Office.Interop.Word.Application app = new Microsoft.Office.Interop.Word.Application();

var Wrd = new Microsoft.Office.Interop.Word.Application();

Wrd.Visible = true;

var inf = Type.Missing;

string str;

var Doc = Wrd.Documents.Add(inf, inf, inf, inf);

Wrd.Selection.TypeText("Исходный массив");

Object t1 = Microsoft.Office.Interop.Word.WdDefaultTableBehavior.wdWord9TableBehavior;

Object t2 = Microsoft.Office.Interop.Word.WdAutoFitBehavior.wdAutoFitContent;

Microsoft.Office.Interop.Word.Table tbl = Wrd.ActiveDocument.Tables.Add(Wrd.Selection.Range, stroki + 1, stolbi + 1, t1, t2);

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

{

tbl.Cell(i + 2, 1).Range.InsertAfter("[" + Convert.ToString(i) + "]");

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

{

str = String.Format("{0:f0}", mas[i, j]);

tbl.Cell(i + 2, j + 2).Range.InsertAfter(str);

}

}

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

tbl.Cell(1, i + 2).Range.InsertAfter("[" + Convert.ToString(i) + "]");

}

public static void ZapisBloknot(int[,] mas, int cols, int rows)

{

StreamWriter rez = File.CreateText("Lr6\_mas.txt"); ;

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

{

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

{

rez.Write(mas[i, j] + "\t");

}

rez.WriteLine("\n");

}

rez.Close();

System.Diagnostics.Process.Start("Lr6\_mas.txt");

}

public static void Zap\_Excel(int[,] mas, int[] rezmas, int cols, int rows)

{

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

Microsoft.Office.Interop.Excel.Workbook workbook = excel.Workbooks.Open(@"C:\Users\Bulat\Desktop\АиП\Exe");

Microsoft.Office.Interop.Excel.Worksheet worksheet;

worksheet = (Microsoft.Office.Interop.Excel.Worksheet)workbook.Worksheets.get\_Item(1);

worksheet.Name = "Массив исходный";

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

{

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

{

worksheet.Cells[i + 1, j + 1] = mas[i, j];

}

}

worksheet.Cells[9, 1] = "Результирующий массив";

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

{

worksheet.Cells[10, i + 1] = "[" + i + "]";

worksheet.Cells[11, i + 1] = rezmas[i];

}

worksheet.Range[("A15")].Select();

excel.Visible = true;

excel.UserControl = true;

}

public static void ZapisWord(int cols, int rows, int[,] mas, int[] rezmas, int g)

{

Microsoft.Office.Interop.Word.Application app = new Microsoft.Office.Interop.Word.Application();

var Wrd = new Microsoft.Office.Interop.Word.Application();

Wrd.Visible = true;

var inf = Type.Missing;

String str;

var Doc = Wrd.Documents.Add(inf, inf, inf, inf);

Wrd.Selection.TypeText("Исходный массив");

Object t1 = Microsoft.Office.Interop.Word.WdDefaultTableBehavior.wdWord8TableBehavior;

Object t2 = Microsoft.Office.Interop.Word.WdAutoFitBehavior.wdAutoFitContent;

Microsoft.Office.Interop.Word.Table tbl = Wrd.ActiveDocument.Tables.Add(Wrd.Selection.Range, 2, cols, t1, t2); //&cols&

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

{

tbl.Cell(i + 2, 1).Range.InsertAfter("[" + Convert.ToString(i) + "]");

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

{

tbl.Cell(1, j + 2).Range.InsertAfter("[" + Convert.ToString(j) + "]");

}

}

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

{

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

{

str = String.Format("{0:f2}", mas[i, j]);

tbl.Cell(i + 2, j + 2).Range.InsertAfter(str);

}

}

Object t3 = Microsoft.Office.Interop.Word.WdUnits.wdLine;

Object add\_str = cols + 2;

Wrd.Selection.MoveDown(t3, add\_str, inf);

Wrd.Selection.TypeText("Количество положительных элементов: " + g + "\nРезультирующий массив");

tbl = Wrd.ActiveDocument.Tables.Add(Wrd.Selection.Range, 2, g, t1, t2);

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

{

tbl.Cell(1, i + 1).Range.InsertAfter("[" + Convert.ToString(i) + "]");

str = String.Format("{0:f2}", rezmas[i]);

tbl.Cell(2, i + 1).Range.InsertAfter(str);

}

}

public static void add()

{

var k = new ADOX.Catalog();

try

{

k.Create("Provider = Microsoft.Jet.OLEDB.4.0; Data Source = C:\\Users\\Bulat\\Desktop\\Мои проекты\\Курсовые\\АиП\\\_Курсовая\_db.accdb");

MessageBox.Show("БД успешно создана", "Информация", MessageBoxButtons.OKCancel, MessageBoxIcon.Information);

}

catch (System.Runtime.InteropServices.COMException sit)

{

MessageBox.Show(sit.Message);

}

finally

{

k = null;

}

}

public static void CreateStructBD(int cols)

{

string commandString = "Provider = Microsoft.Jet.OLEDB.4.0; Data Source = C:\\Users\\Bulat\\Desktop\\Мои проекты\\Курсовые\\АиП\\\_Курсовая\_db.accdb";

var Connect = new OleDbConnection(commandString);

Connect.Open();

var Command = new OleDbCommand();

Command.Connection = Connect;

string Col = $@"CREATE TABLE Massivs (№ CHAR(100), ";

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

{

if(i == cols-1)

{

Col = Col + $"Column\_{i + 1} CHAR(100))";

}

else

{

Col = Col + $"Column\_{i + 1} CHAR(100), ";

}

}

Command.CommandText = Col;

try

{

Command.ExecuteNonQuery();

MessageBox.Show("Структура БД создана", "Информация", MessageBoxButtons.OKCancel);

}

catch (System.Runtime.InteropServices.COMException sit)

{

MessageBox.Show(sit.Message);

}

Connect.Close();

}

public static void CommBD(string commandString, OleDbConnection conn)

{

conn.Open();

//var Connection = new OleDbConnection();

//Connection.Open();

var Command = new OleDbCommand(commandString);

Command.Connection = conn;

Command.ExecuteNonQuery();

conn.Close();

}

public static void AddToBD(int cols, int[,] mas, int rows)

{

string cmdString;

string cmdString2;

var conn = new OleDbConnection("Provider = Microsoft.Jet.OLEDB.4.0; DataSource = C:\\Users\\Bulat\\Desktop\\Мои проекты\\Курсовые\\АиП\\\_Курсовая\_db.accdb");

conn.Open();

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

{

cmdString = "INSERT INTO Massive (№";

cmdString2 = ") VALUES (" + "'" + Convert.ToString(i + 1) + "'";

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

{

cmdString = cmdString + ", [Col" + Convert.ToString(j + 1) + "]";

cmdString2 = cmdString2 + ",'" + Convert.ToString(mas[i,j]) + "'";

}

cmdString = cmdString + ")";

cmdString2 = cmdString2 + ")";

CommBD(cmdString + cmdString2, conn);

//Connect.Close();

}

MessageBox.Show("Запись в базу данных внесена");

}

}

}

1. **Результат выполнения проекта**

Результат работы программы представлен на рисунке 2.

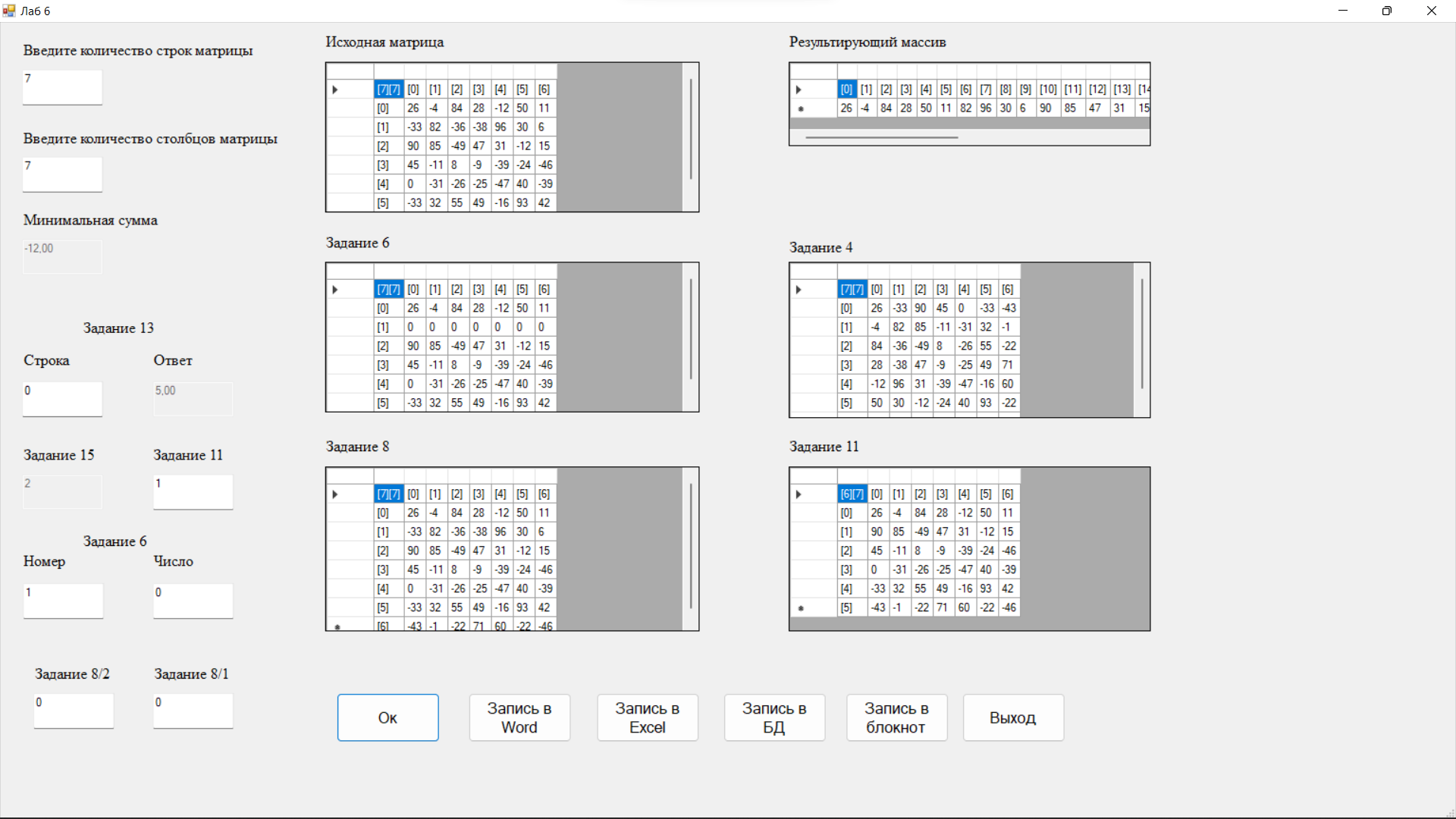


Рисунок 2 - Результат выполнения проекта

Результат создание базы данных для двумерных массивов представлен на рисунке 3.

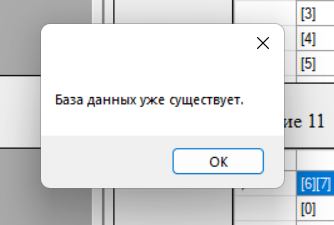


Рисунок 3 – Создание базы данных

Результат записи структуры базы данных для двумерных массивов представлен на рисунке 4.

Рисунок 4 – Создание структуры базы данных

Результат записи двумерных массивов в базу данных представлен на рисунках 5-6.

Рисунок 5 – Запись массивов в базу данных

Рисунок 6 – Интерфейс Microsoft Access DataBase

Результат записи исходного массива в Блокнот представлен на рисунке 7.

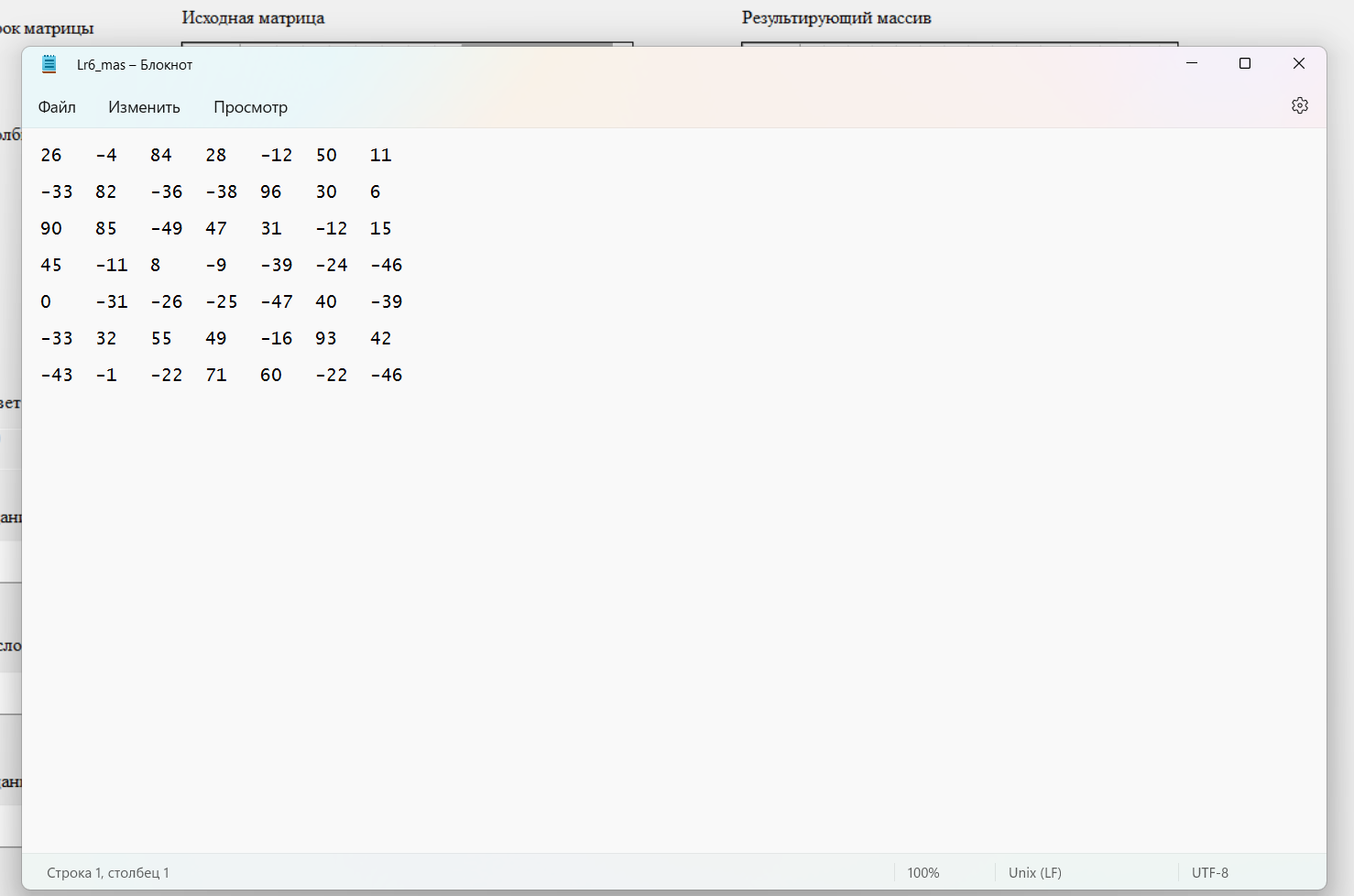


Рисунок 7 – Интерфейс Блокнот

Результат записи исходного массива в Microsoft Excel представлен на рисунке 8.

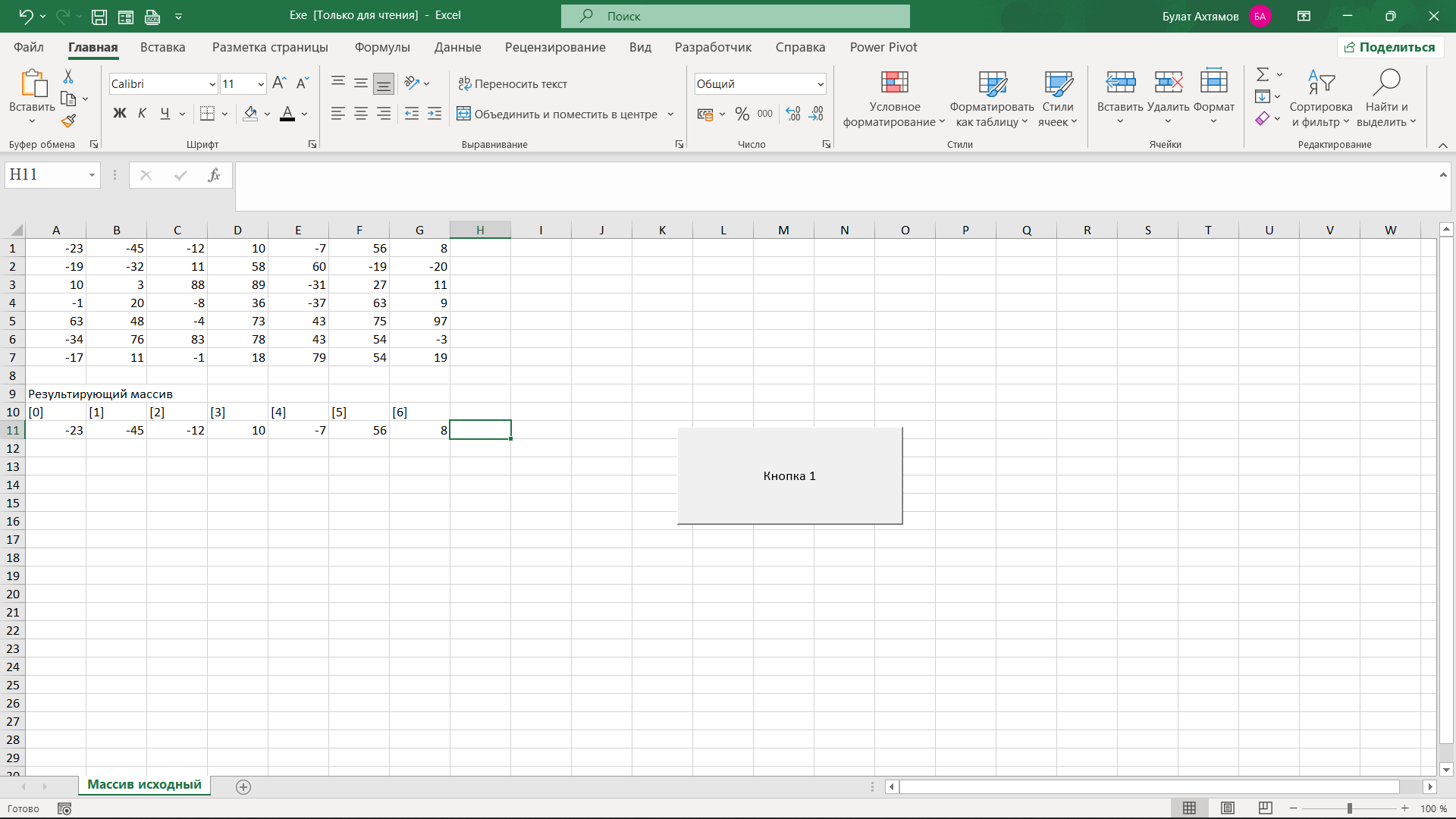


Рисунок 8 – Интерфейс Microsoft Excel

Результат выполнения макроса в Microsoft Excel представлен на рисунке 9.

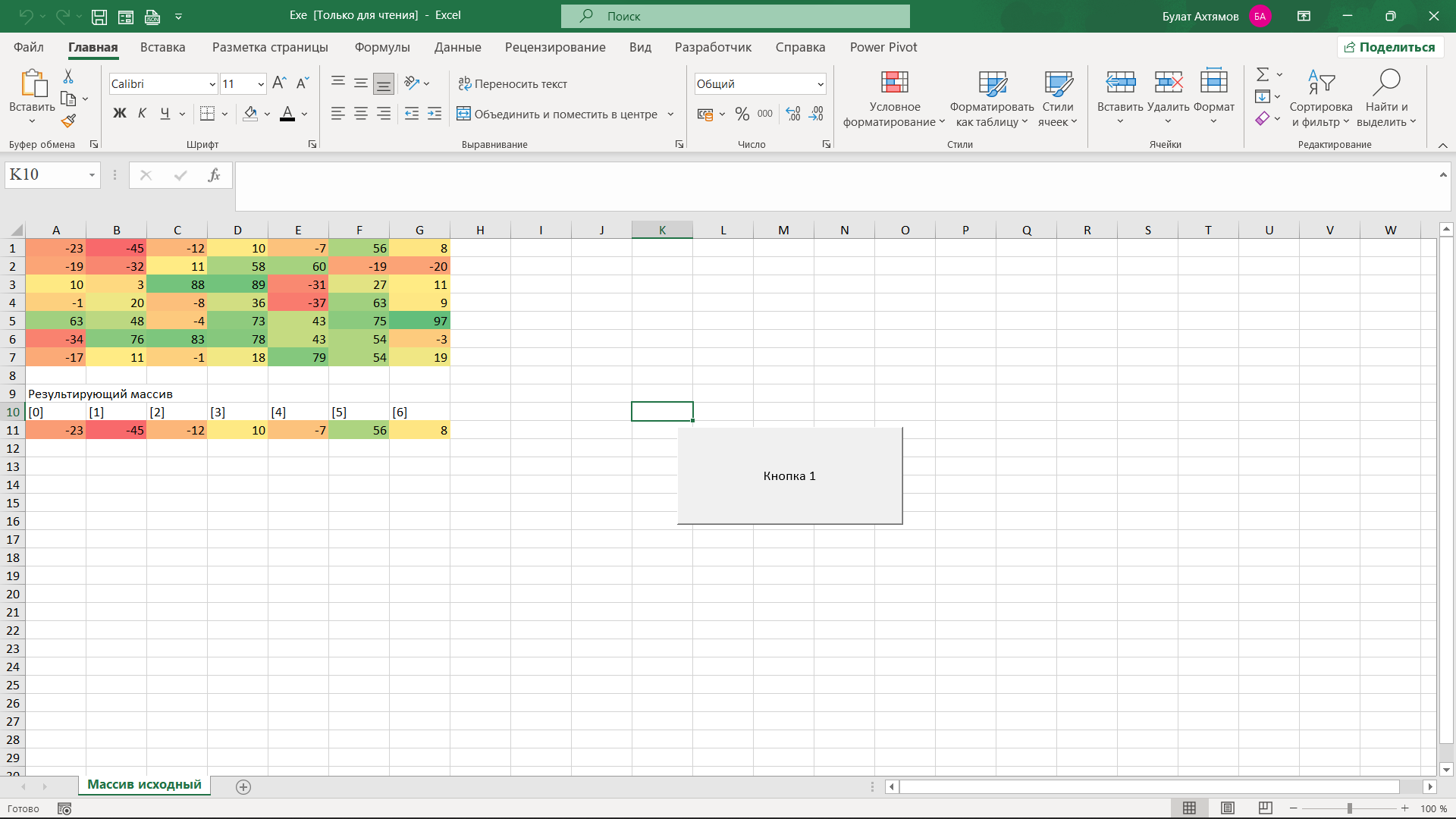


Рисунок 9 – макроса в Microsoft Excel

Результат записи исходного массива в Microsoft Word представлен на рисунке 10.

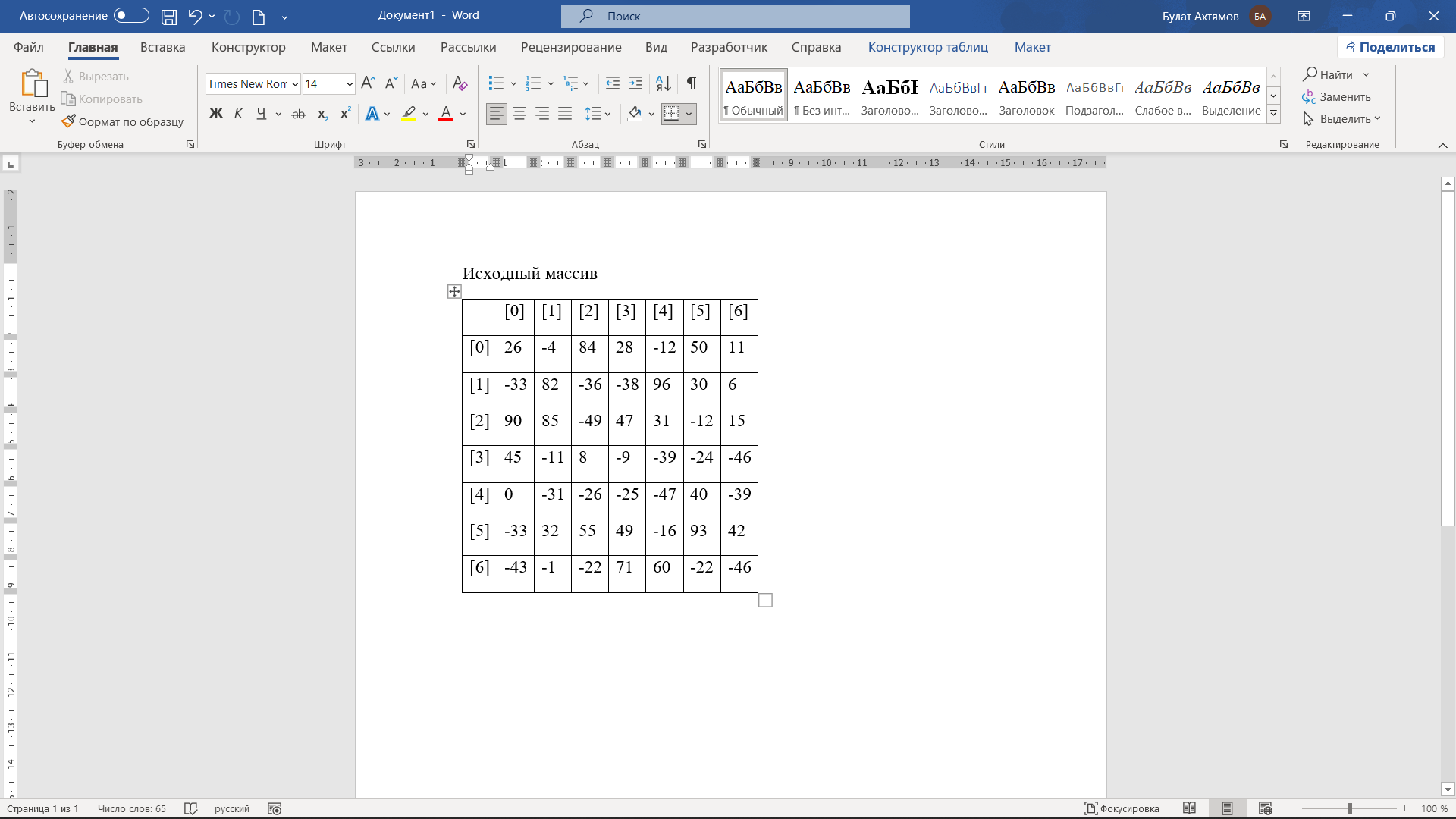


Рисунок 10 – Интерфейс Microsoft Word