**Лабораторная работа № 4**

**«ОБРАБОТКА ДВУМЕРНЫХ МАССИВОВ»**

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. 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.

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

Разработанная форма интерфейса пользователя представлена на рисунке 1.

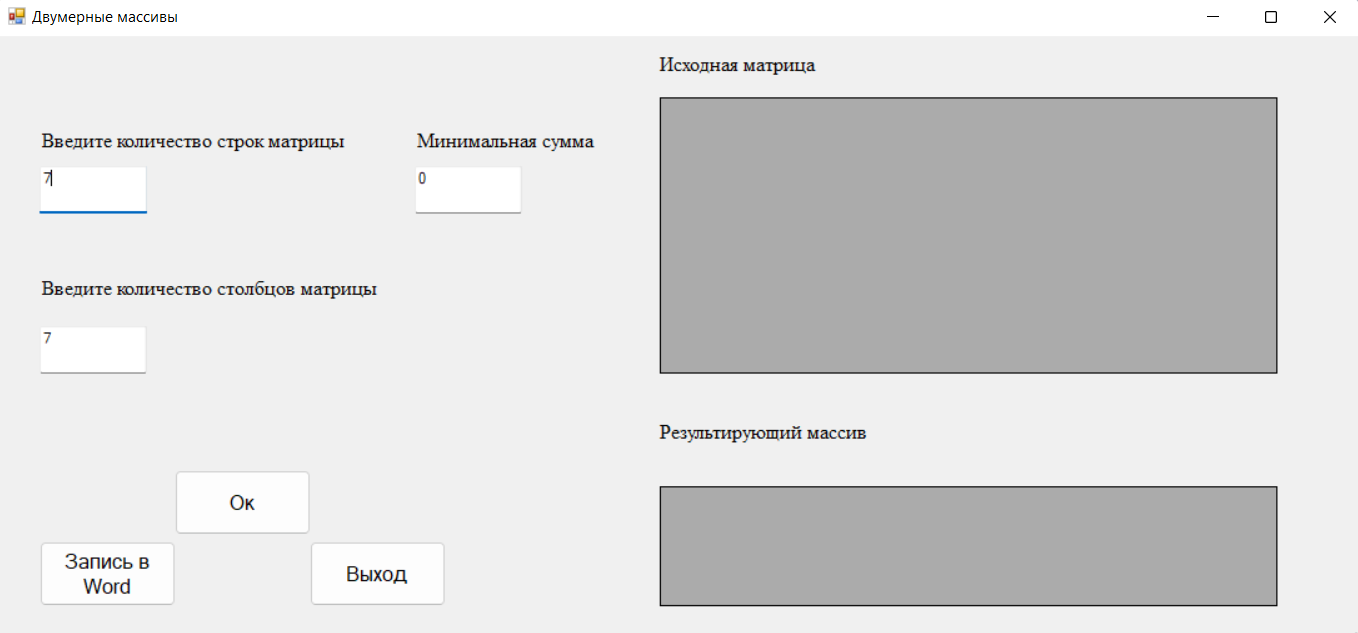


Рисунок 1 - Графический интерфейс пользователя

* 1. Разработка схемы алгоритмов

Блок-схемы событийной части проекта представлен на рисунках 2 - 3.

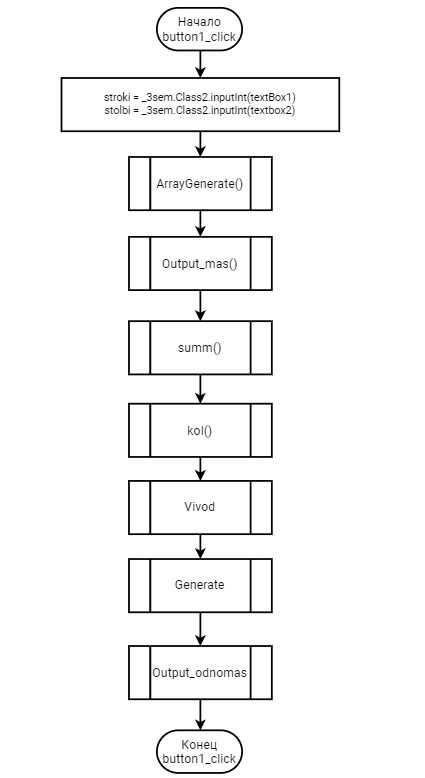


Рисунок 2 - Схема алгоритмов процедур **button\_Click1**

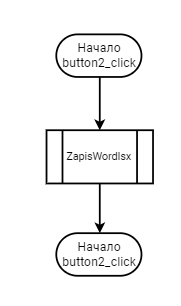


Рисунок 3 - Схема алгоритмов процедур **button\_Click2**

Проверка ввода данных будет применяться на все элементы управления ***Textbox***. Блок-схемы представлен на рисунке 4.

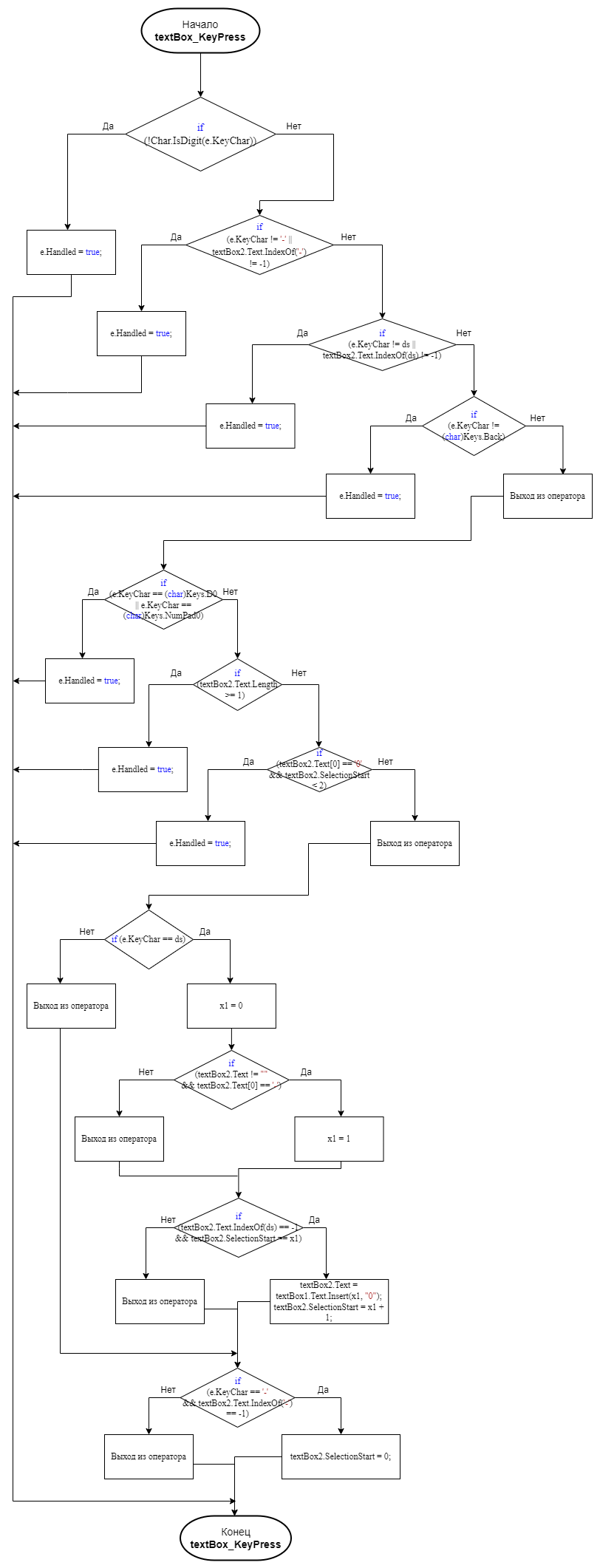


Рисунок 4 - Схема алгоритмов процедур **textbox\_KeyPress**

Блок-схем функций, подключенных из библиотеки ***Class2***, представлен на рисунках 5 – 8.

Блок-схема генерации массива представлена на рисунке 5.

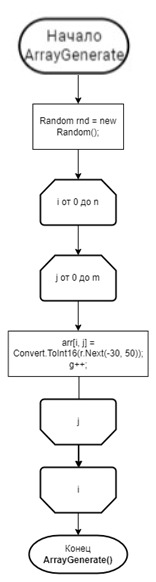


Рисунок 5 - Схема алгоритмов процедур **ArrayGenerate()**

Блок-схема программы, рассчитывающей минимальную сумму положительных значений по главной и побочной диагонали.

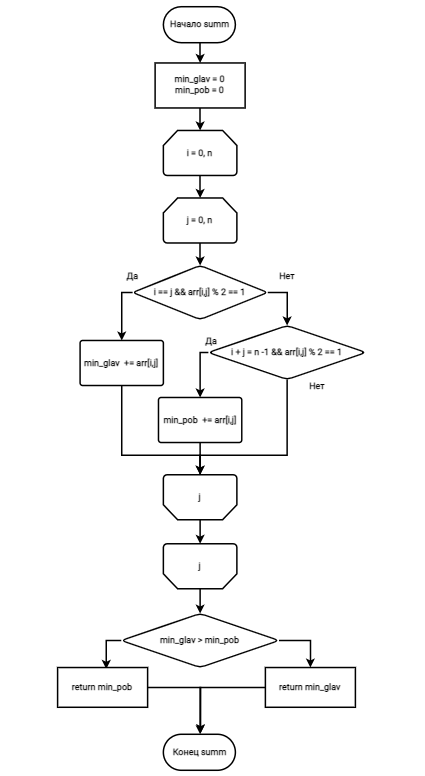


Рисунок 6 - Схема алгоритмов процедур **min()**

Блок-схема, предназначенный для вывода двумерного массива в таблицу представлена на рисунке 7.

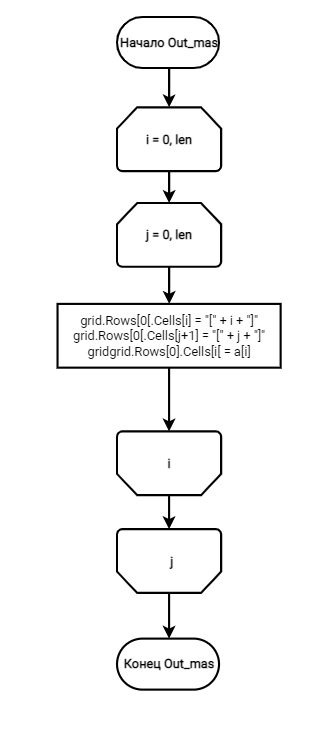


Рисунок 7 – Схема алгоритмов для вывода двумерного массива

Блок-схема, предназначенная для вывода одномерного массива в таблицу представлена на рисунке 8.

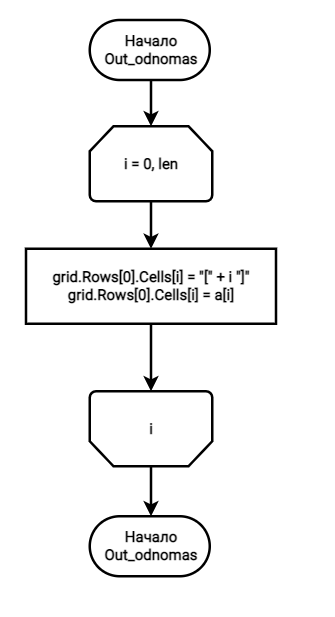
****

Рисунок 8 - Схема алгоритмов процедуры **Out\_odnommas()**

* 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.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AnP\_proj

{

public partial class Form12 : Form

{

public Form12()

{

InitializeComponent();

}

public static int[,] arr;

public static int[] rezmas;

public static int stroki;

public static int stolbi;

public static int length;

public int min;

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

{

dataGridView1.Rows.Clear();

dataGridView2.Rows.Clear();

dataGridView1.ColumnCount = 0;

dataGridView2.ColumnCount = 0;

//vvod parametrov

stroki = \_3sem.Class2.inputInt(textBox1);

stolbi = \_3sem.Class2.inputInt(textBox2);

arr = new int[stolbi, stroki];

\_3sem.Class2.ArrayGenerate(ref arr, stolbi, stroki);

\_3sem.Class2.Output\_mas(stolbi, stroki, ref arr, ref dataGridView1);

\_3sem.Class2.summ(ref arr, stolbi, stroki, ref min);

length = \_3sem.Class2.Kol(ref arr, stolbi, stroki, min);

rezmas = new int[length];

textBox3.Text = Convert.ToString(min);

\_3sem.Class2.Generate(ref arr, ref rezmas,stolbi, stroki, min);

\_3sem.Class2.Output\_odnomas(ref dataGridView2, length, rezmas);

}

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

{

Application.Exit();

}

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

{

\_3sem.Class2.ZapisWordIsx(length, rezmas);

}

}

}

Все подпрограммы-методы размещаются в классе ***Class2***.

u 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

{

//generation array

public static void ArrayGenerate(ref int[,] arr, int n, int m)

{

Random rnd = new Random();

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

{

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

{

arr[i, j] = Convert.ToInt16(rnd.Next(-50,100));

}

}

}

// output array

public static void Output\_mas(int n, int m, ref int[,] arr, ref DataGridView grid1)

{

grid1.ColumnCount = m + 1;

grid1.RowCount = n + 1;

grid1.Rows[0].Cells[0].Value = "[" + n + "]" + "[" + m + "]";

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

{

grid1.Rows[i+1].Cells[0].Value = "[" + i + "]";

}

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

{

grid1.Rows[0].Cells[j+1].Value = "[" + j + "]";

}

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

{

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

{

grid1.Rows[i+1].Cells[j+1].Value = arr[i, j];

}

}

}

//output Array with condition

public static void Output\_odnomas(ref DataGridView grid, int len, params int[] a)

{

grid.ColumnCount = len;

grid.RowCount = 2;

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

{

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

grid.Rows[1].Cells[i].Value = a[i];

}

}

//output count of array with condition

public static int Kol(ref int[,] arr, int n, int m, int suma\_glav)

{

int count = 0;

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

{

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

{

if (arr[i, j] > suma\_glav)

{

count++;

}

}

}

return count;

}

//doit arr

public static void Generate(ref int[,] arr, ref int[] rezmas, int n, int m, int suma\_glav)

{

int count = 0;

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

{

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

{

if (arr[i, j] > suma\_glav)

{

rezmas[count] = arr[i, j];

count++;

}

}

}

}

public static void summ(ref int[,] arr, int n, int m, ref int min)

{

int min\_glav = 0;

int min\_pob = 0;

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

{

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

{

if ((i == j) && (Math.Abs(arr[i,j]) % 2 == 1))

{

min\_glav += arr[i, j];

}

if ((i + j == n-1) && (Math.Abs(arr[i, j]) % 2 == 1))

{

min\_pob += arr[i, j];

}

}

}

if (min\_glav > min\_pob)

{

min = min\_pob;

}

else

{

min = min\_glav;

}

}

//generation of array with condition

public static void Set\_rezmas(ref int[,] arr, ref int[] rezarr,int n, int m, int k, ref int g)

{

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

{

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

{

if(arr[i, j] >= k)

{

rezarr[g++] = arr[i, j];

}

}

}

}

public static void ZapisWordIsx(int length, params 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.wdWord8TableBehavior;

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

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

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

{

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

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

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

}

}

public static int inputInt(TextBox t)

{

if (Convert.ToString(t.Text) != "")

{

if (Convert.ToInt32(t.Text) > 0)

{

return Convert.ToInt32(t.Text);

}

else { return 0; }

}

else { return 0; }

}

}

}

1. **Результат выполнения проекта**

Результат работы программы представлен на рисунке 10.

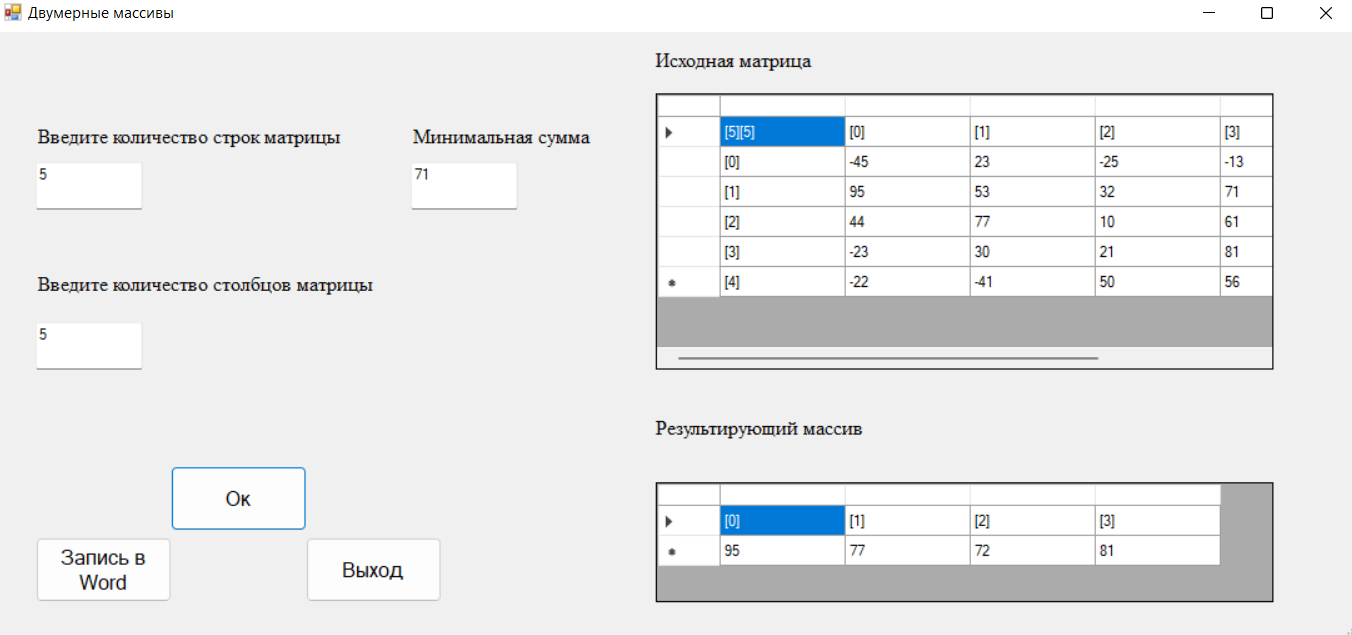


Рисунок 10 - Результат выполнения проекта

Вывод одномерного результирующего массива в Microsoft Word представлен на рисунке 11.

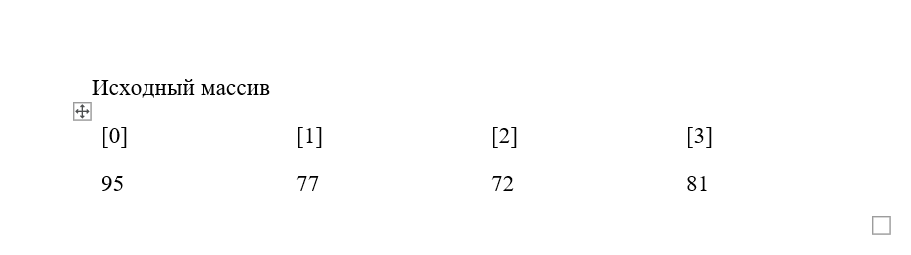


Рисунок 11 - Вывод массива в Microsoft Word