**Лабораторная работа №8**

**“Базовые алгоритмы обработки и методы сортировки одномерных массивов”**

Дан динамический целочисленный массив. Опишите алгоритм, позволяющий найти и вывести наименьшую нечётную сумму двух соседних элементов массива.

Разработайте метод, который формирует новый массив, из тех элементов исходного массива, значения которых больше найденного числа.

Кроме того, необходимо реализовать алгоритм, позволяющий найти и вывести сумму всех трёхзначных чисел, десятичная запись которых оканчивается на 9, но не на 99, если их нет вывести -1. После отсортировать массив двумя видами сортировок простой вставкой и простым обменом.

Для выполнения задания следует разобрать нижеперечисленные методы:

* Метод ***Vvod()***, предназначенный для ввода данных;
* Метод ***Vivod()***, предназначенный для вывода данных;
* Метод ***arrayGenerate()***, предназначенный для создания массива, состоящего из случайного набора чисел;
* Метод ***output\_mas()***, предназначенный для вывода массива в таблицу;
* Метод ***set\_mas()***, формирующий новый массив, из тех элементов исходного массива, которые больше наименьшей нечетной суммы соседних элементов;
* Метод ***poisk()***, созданный для поиска пары соседних чисел с наименьшей нечётной суммой;
* Метод ***VivodDGV***(), созданный для вывода значений больших наименьшей суммы с нечётным значением;
* Метод ***InputInt()***, выводит число из ***textbox***;
* Метод ***Sort\_prost***, сортирующая простым методом;
* Метод ***Sort\_obmen***, сортирующая обменом;
* Метод ***Zad\_3***, выполняющая задание, описанное выше;

**2. Разработка проекта**

2.1. Разработка графического интерфейса

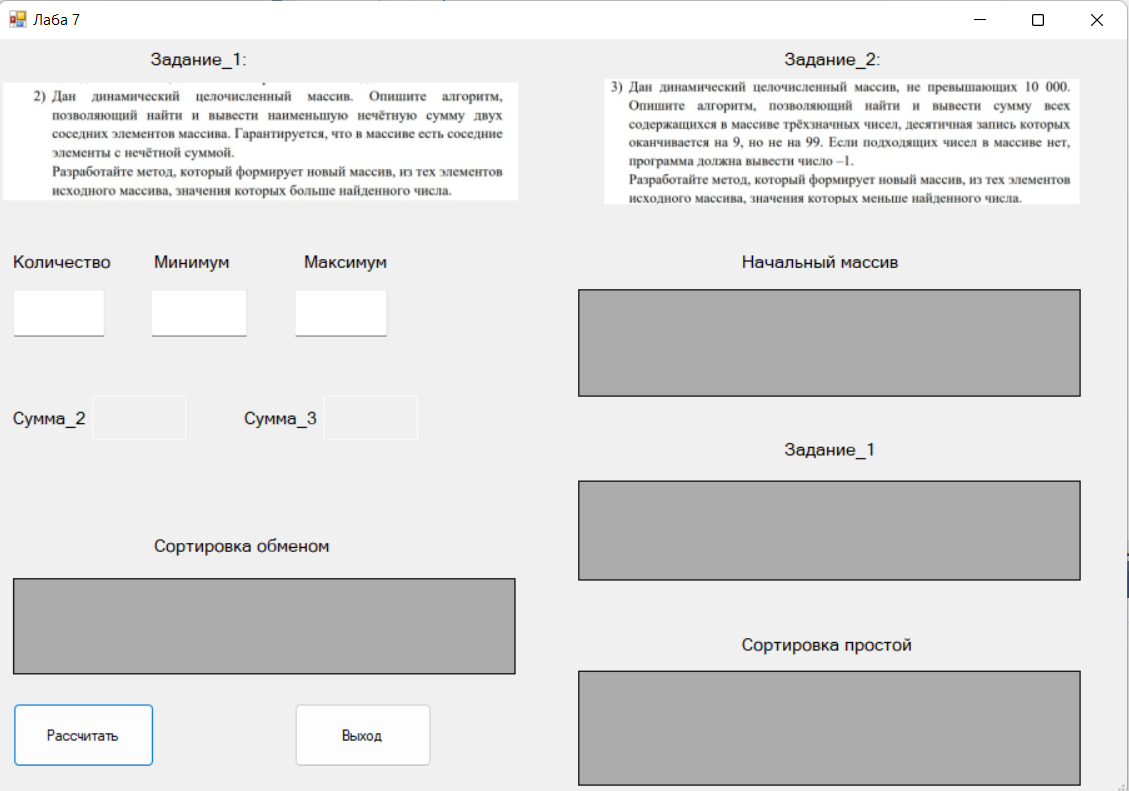


Рисунок 1 – Графический интерфейс пользователя

2.2. Разработка проекта схемы алгоритмов

Блок схемы событийной части проекта представлены на рисунках 2-3

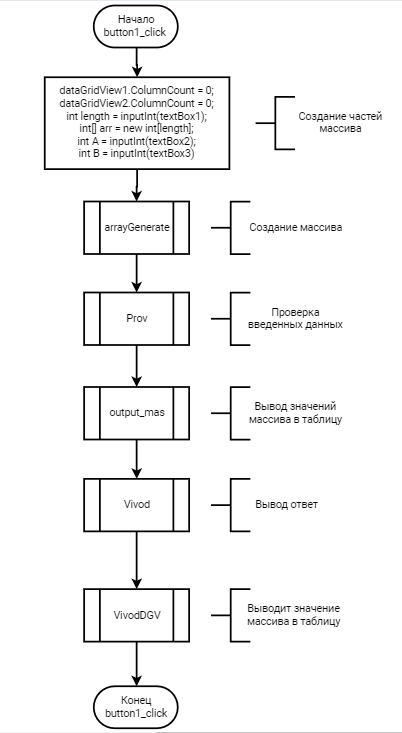


Рисунок 2 - Схема алгоритмов процедур **button\_Click1**

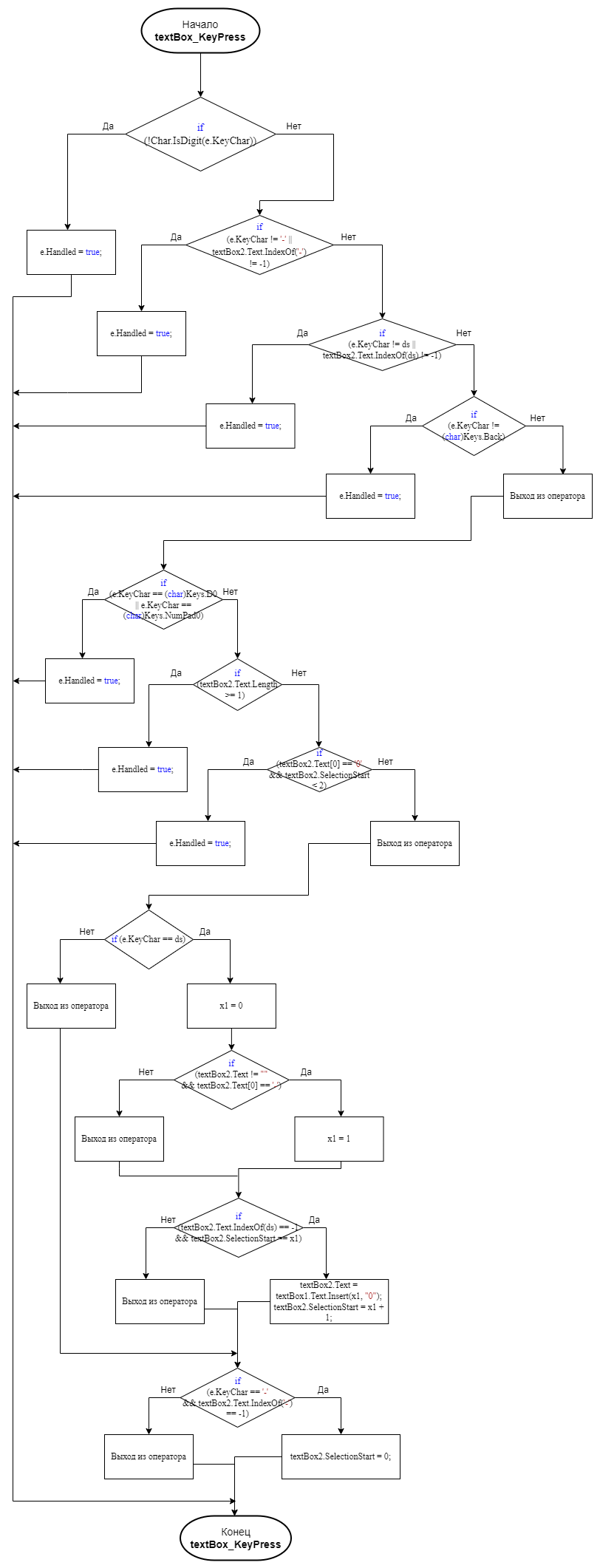


Рисунок 3 – схема алгоритмов процедур KeyPress

Блок схемы функций представлены на рисунках 4-10.

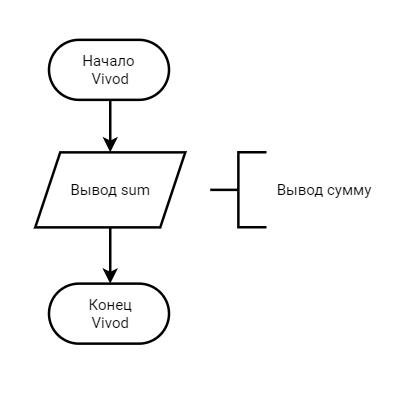


Рисунок 4 - Схема алгоритмов процедур **Vivod**

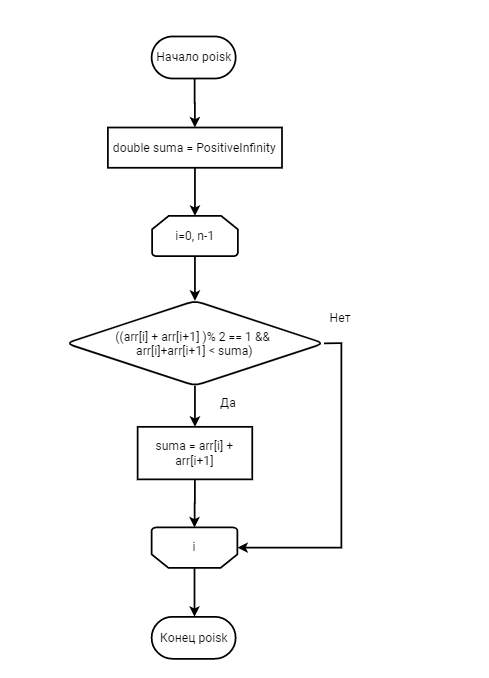


Рисунок 5 - Схема алгоритмов процедур **poisk**

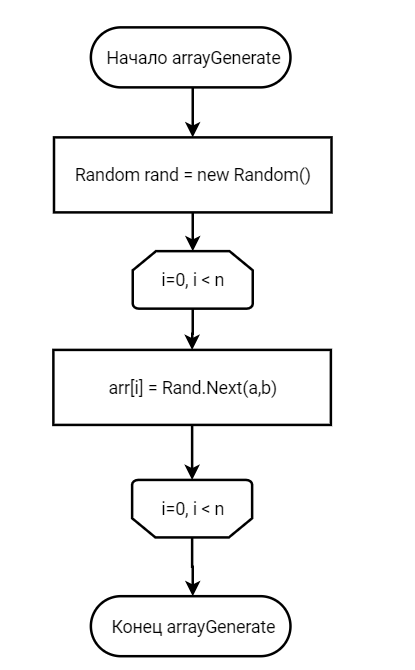


Рисунок 6 - Схема алгоритмов процедур **arrayGenerate**

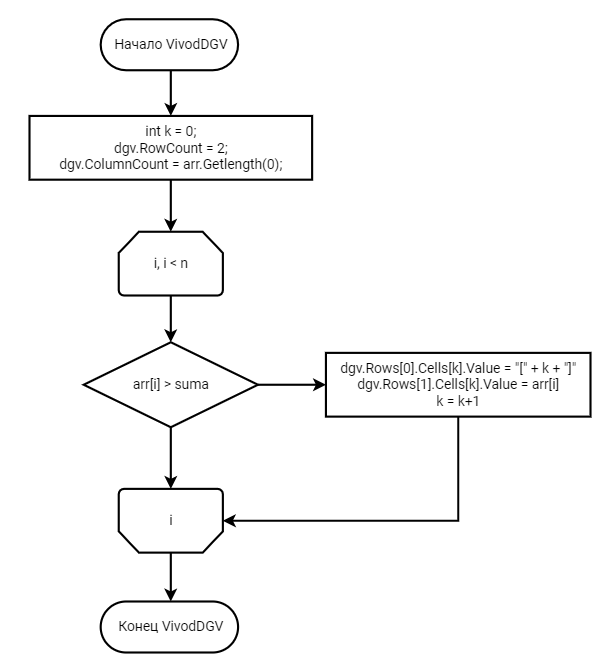
****

Рисунок 7 - Схема алгоритмов процедур **VivodDGV**

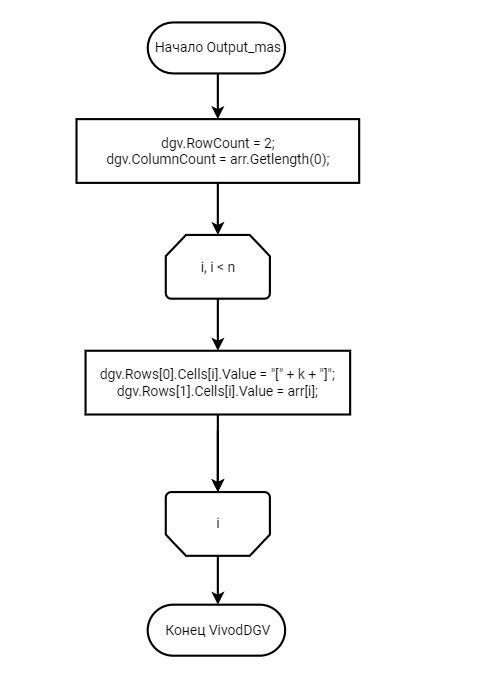
****

Рисунок 10 - Схема алгоритмов процедур **Output\_mas**

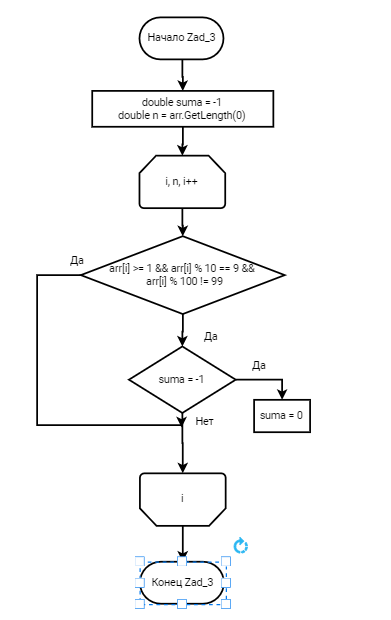
****

Рисунок 11 - Схема алгоритмов процедур **Zad\_3**

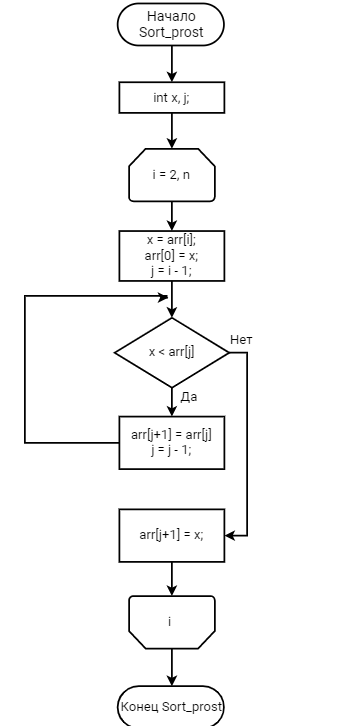
****

Рисунок 11 - Схема алгоритмов процедур **Sort\_prost**

2.3. Написание программного кода

Основной код

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 Form8 : Form

{

public Form8()

{

InitializeComponent();

}

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

{

dataGridView1.Rows.Clear();

dataGridView2.Rows.Clear();

dataGridView1.ColumnCount = 0;

dataGridView2.ColumnCount = 0;

//vvod parametrov

int length = Laba\_7.Class1.inputInt(textBox1); //count

if (length <= 0) return;

//generation and print array

int[] arr = new int[length];

int A = Laba\_7.Class1.inputInt(textBox2); //start

int B = Laba\_7.Class1.inputInt(textBox3); //stop

Laba\_7.Class1.arrayGenerate(ref arr, length, A, B);

//Laba\_7.Class1.arrayOutput(ref arr, lenght, ref dataGridView1);

Laba\_7.Class1.output\_mas(ref dataGridView1, arr);

double suma = Laba\_7.Class1.poisk(arr);

Laba\_7.Class1.Vivod(textBox4, suma);

Laba\_7.Class1.VivodDGV(arr, suma, dataGridView2);

double suma\_2 = Laba\_7.Class1.Zad\_3(arr);

Laba\_7.Class1.Vivod(textBox5, suma\_2);

//Laba\_7.Class1.Sort\_prost(ref arr);

//Laba\_7.Class1.output\_mas(ref dataGridView3, arr);

Laba\_7.Class1.Sort\_obmen(ref arr);

Laba\_7.Class1.output\_mas(ref dataGridView4, arr);

}

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

{

Application.Exit();

}

private void textBox3\_KeyPress(object sender, KeyPressEventArgs e)

{

Laba\_7.Class1.Prof(textBox3, e);

}

private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

Laba\_7.Class1.Prof(textBox2, e);

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

Laba\_7.Class1.Prof(textBox1, e);

}

}

}

Код из библиотеки class1

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;

namespace Laba\_7

{

public class Class1

{

public static double poisk(int[] arr)

{

double suma = double.PositiveInfinity;

for(int i = 0; i < arr.Length - 1; i++)

{

if((arr[i] + arr[i+1])%2 == 1)

{

if (arr[i] + arr[i + 1] < suma)

{

suma = arr[i] + arr[i + 1];

}

}

}

return suma;

}

public static double VivodDGV(int[] arr, double suma, DataGridView dgv)

{

int k = 0;

dgv.RowCount = 2;

dgv.ColumnCount = arr.GetLength(0);

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

{

if (arr[i] > suma)

{

dgv.Rows[0].Cells[k].Value = "[" + k + "]";

dgv.Rows[1].Cells[k].Value = arr[i];

k = k + 1;

}

}

return 0;

}

public static void arrayGenerate(ref int[] arr, int length, int a, int b)

{

Random rand = new Random();

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

{

arr[i] = rand.Next(a, b);

}

}

public static void output\_mas(ref DataGridView dgv, params int[] a)

{

dgv.RowCount = 2;

dgv.ColumnCount = a.GetLength(0);

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

{

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

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

}

}

public static void set\_mas(int k, ref int[] rezmas, ref int j, params int[] mas)

{

j = 0;

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

{

if (mas[i] > k) // if znac more than int

{

rezmas[i] = mas[i];

j++;

}

}

}

public static int inputInt(TextBox t)

{

return Convert.ToInt32(t.Text);

}

public static void Vivod(TextBox t, double c)

{

t.Text = c.ToString("F");

}

public static void Sort\_prost(ref int[] arr)

{

int x, j;

for (int i = 2; i < arr.GetLength(0); i++)

{

x = arr[i];

arr[0] = x;

j = i - 1;

while (x < arr[j])

{

arr[j + 1] = arr[j];

j = j - 1;

}

arr[j + 1] = x;

}

}

public static void Sort\_obmen(ref int[] arr)

{

int x;

int n = arr.GetLength(0);

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

{

for (int j = n; j < i; j = j-1)

{

if (arr[j-1] > arr[j])

{

x = arr[j - 1];

arr[j-1] = arr[j];

arr[j] = x;

}

}

}

}

public static double Zad\_3(int[] arr)

{

double suma = -1;

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

{

if ((arr[i] / 100 >= 1) && (arr[i] % 10 == 9) && (arr[i] % 100 != 99))

{

if (suma == -1)

{

suma = 0;

}

suma = suma + arr[i];

}

}

return suma;

}

public static void Prof(TextBox textBox1, KeyPressEventArgs e)

{

int x1;

char ds = (char)NumberFormatInfo.CurrentInfo.NumberDecimalSeparator[0];

// Если нажатая клавиша не цифровая.

if (!Char.IsDigit(e.KeyChar))

// Запрет на ввод более одного знака минуса.

if (e.KeyChar != '-' || textBox1.Text.IndexOf('-') != -1)

// Запрет на ввод более одного десятичного разделителя.

if (e.KeyChar != ds || textBox1.Text.IndexOf(ds) != -1)

// Если нажатая клавиша не является клавишей BackSpace.

if (e.KeyChar != (char)Keys.Back)

e.Handled = true; // Запрет ввода

// Запрет ввода повторных нулей в начале числа.

if (e.KeyChar == (char)Keys.D0 || e.KeyChar == (char)Keys.NumPad0)

if (textBox1.Text.Length >= 1) // Если строка не пустая

if (textBox1.Text[0] == '0' && textBox1.SelectionStart < 2)

e.Handled = true; // Запрет ввода

//-незначащий ноль

if (Char.IsDigit(e.KeyChar))

if (textBox1.Text.Length == 1 && textBox1.Text[0] == '0')

textBox1.Paste(",");

// Замена десятичного разделителя в начале числа на "0,".

if (e.KeyChar == ds)

{ // Если введен десятичный разделитель

x1 = 0;

if (textBox1.Text != "" && textBox1.Text[0] == '-') { x1 = 1; }

if (textBox1.Text.IndexOf(ds) == -1 && textBox1.SelectionStart == x1)

{

// Если десятичный разделитель уже есть, то 0 не вставлять.

textBox1.Text = textBox1.Text.Insert(x1, "0"); // Вставка 0

textBox1.SelectionStart = x1 + 1; // Текстовый курсор в конец

}

// Перенос точки ввода минуса в начало (ввод минуса только спереди)

}

if (e.KeyChar == '-' && textBox1.Text.IndexOf('-') == -1)

textBox1.SelectionStart = 0; // Текстовый курсор в начало

//-незначащий ноль

}

}

}

**3. Результат выполнение проекта**

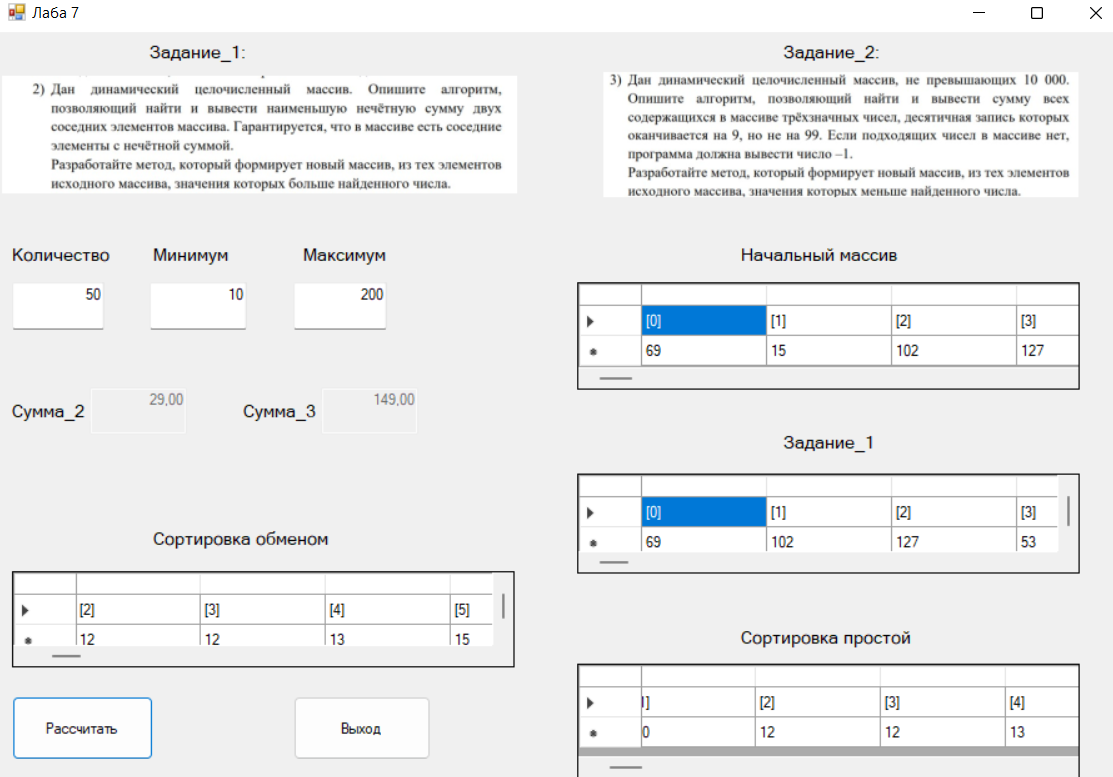
****

Рисунок 13 – Результат выполнение программы