

**КАЗАНСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ  
ИНСТИТУТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ  
И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

*Кафедра системного анализа и информационных технологий*

**Б.Р. АБАЙДУЛЛИН, Р.Р. ТАГИРОВ**

**ЛАБОРАТОРНЫЕ РАБОТЫ ПО КУРСУ  
ОСНОВЫ ПРОГРАММИРОВАНИЯ  
(языки Си# и Си++)**

**Учебно-методическое пособие**

**Казань – 2017**

УДК 004.43  
ББК 32.973.26 - 018.1

*Принято на заседании учебно-методической комиссии  
Института вычислительной математики и информационных технологий  
Протокол № 9 от 4 мая 2017 года*

*заседании кафедры системного анализа и информационных технологий  
Протокол № 7 от 5 апреля 2017 года*

**Рецензенты:**

доктор технических наук, профессор кафедры высшей математики института  
управления автоматизации и информационных технологий КНИТУ

**Е.К. Вачагина;**

кандидат технических наук, старший преподаватель кафедры  
радиоэлектроники института физики КФУ **Е.А. Марфин**

**Абайдуллин Б.Р., Тагиров Р.Р.**

**Лабораторные работы по курсу Основы программирования:** учеб.-  
метод. пособие / Б.Р. Абайдуллин, Р.Р. Тагиров. - Казань: Казан. ун-т, 2017. -  
73 с.

В данном пособии описываются основные инструменты и алгоритмы языка программирования Си# и Си++ при решении конкретных задач. Пособие представляет собой перечень тем и задач и предназначено для студентов, обучающихся по направлениям «Прикладная математика и информатика», «Информационная безопасность», и «Фундаментальная информатика и информационные технологии».

© Абайдуллин Б.Р., Тагиров Р.Р., 2017

© Казанский университет, 2017

## **СОДЕРЖАНИЕ**

ВВЕДЕНИЕ .....	4
ТЕМА 1: ОСНОВНЫЕ ЭЛЕМЕНТАРНЫЕ АЛГОРИТМЫ.....	4
ТЕМА 2: ГЕОМЕТРИЯ.....	16
ТЕМА 3: ПОСЛЕДОВАТЕЛЬНОСТИ.....	24
ТЕМА 4: ОДНОМЕРНЫЕ МАССИВЫ.....	30
ТЕМА 5: СОРТИРОВКИ.....	36
ТЕМА 6: СИМВОЛЬНЫЕ СТРОКИ (ТЕКСТЫ).....	47
ТЕМА 7: ДВУМЕРНЫЕ МАССИВЫ (МАТРИЦЫ).....	53
ТЕМА 8: ФУНКЦИИ И РЕКУРСИИ .....	65
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	71

## **ВВЕДЕНИЕ**

Одной из важнейших задач при освоении курса фундаментальной информатики является освоение основ программирования на примере какого-либо широко распространенного языка программирования. В связи с широким распространением операционных систем Windows, и соответственно приложений написанных на языке Си++ и Си# (связанных с платформой .NET), акцент в данном пособии сделан на языке Си# и Си++.

Данное учебно-методическое пособие призвано помочь студенту изучить основные конструкции языка программирования, научить студента строить алгоритмы для решения конкретной задачи и реализовать эти алгоритмы на конкретном языке программирования. Пособие содержит как классические задачи по программированию, так и задачи, взятые с сайта по спортивному программированию. Оно адресовано студентам, обучающимся по образовательным программам бакалавриата и магистратуры направлений «Фундаментальная информатика и информационные технологии», «Информационная безопасность», «Прикладная математика и информатика», «Прикладная информатика», реализуемым на кафедре системного анализа и информационных технологий в Институте Вычислительной математики и информационных технологий Казанского (Приволжского) федерального университета.

## **ТЕМА 1: ОСНОВНЫЕ ЭЛЕМЕНТАРНЫЕ АЛГОРИТМЫ**

### **Цель.**

Изучение основных конструкций языка программирования. Умение построить алгоритм для решения задачи и записать его на языке программирования.

### **Основные понятия.**

Решение задачи состоит в описании объектов и указания действий над ними с целью получения нужного результата.

Основной метод решения сложных задач - разбиение задачи на подзадачи - декомпозиция

Существует 3 механизма декомпозиции:

- линейная последовательность – задача разбивается на последовательность шагов (подзадач), которые выполняются друг за другом;
- ветвление по условию – задача разбивается на 2 или несколько параллельных подзадач и в зависимости от некоторого условия будет выполняться только одна из подзадач;
- многократное повторение – задача разбивается на последовательность одинаковых (подобных) подзадач. Эта подзадача будет выполняться несколько раз.

Эти 3 механизма используются при декомпозиции многократно до тех пор,

пока задача не сведётся к очевидным (элементарным) действиям.

В каждой задаче (подзадаче) нужно уметь выделять объекты, над которыми будут выполняться действия.

Объекты характеризуются своим типом и в каждый момент времени имеют некоторое значение, которое хранится в оперативной памяти.

Действия задаются операторами и состоят в выполнении некоторых допустимых операций над объектами.

Структура простой программы:

подключения стандартных библиотек (#include)

определение главной функции (main), либо функция связанная с событием (button\_click()). Внутри неё описываются объекты и операторы.

Ввод данных и печать результатов.

### **Ключевые слова.**

Переменная, константа, Описание. Оператор. Программа. Оператор препроцессора #include.

### **Задача1. Делители числа N.**

Дано натуральное число. Напечатать все его делители в порядке возрастания.

### **Алгоритм.**

Для решения этой задачи будем перебирать все натуральные числа от 1 до самого заданного числа. Те числа, на которые будет делиться без остатка заданное число, будем печатать.

### **Код программы на языке Си++ (консольное приложение Win32).**

```
#include "stdafx.h"
#include <stdio.h>
#include <iostream>

using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{ setlocale(LC_ALL,"Russian");
  int N=0;

  cout<<"Введите N=";
      cin>>N;
      //String^ h = "";
      cout<<"Делители числа N="<<N<<" это:";
      for (int i = 1; i <= N / 2; i++)
      {
          if (N % i == 0)
          {
              cout<<i<<" ";
          }
      }
  }
```

```

        //h += N;
        cout<<N<<" ";
system("pause");
return 0;
}

```

### Визуализация работы программы.

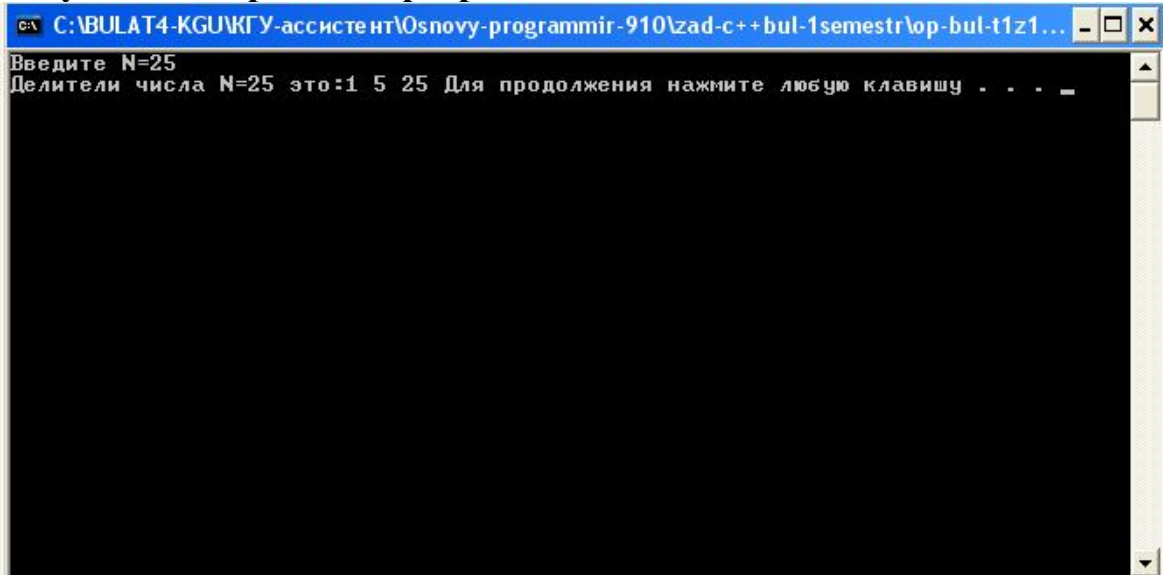


Рис. 1. Визуализация работы программы делителя числа.

### Задача2. Совместимость типов.

Даны целые и вещественные числа  $x$  и  $y$ . Написать программу для преобразования одного типа в другой.

### Код программы на языке Си++ (приложение Windows Forms).

В данном задании код программы демонстрируется полностью, в дальнейшем стандартная часть кода между `#pragma once` и `#pragma endregion` для экономии места и из-за бессмысленности его повторения будем пропускать

```
#pragma once
```

```
namespace bnbnnbnnb {
```

```

    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;

```

```

    enum month { Январь = 1, Февраль, Март, Апрель, Май, Июнь, Июль, Август,
    Сентябрь, Октябрь, Ноябрь, Декабрь };

```

```

/// <summary>
/// Сводка для Form1

```

```

///
/// Внимание! При изменении имени этого класса необходимо также изменить
/// свойство имени файла ресурсов ("Resource File Name") для средства
компиляции управляемого ресурса,
/// связанного со всеми файлами с расширением .resx, от которых
зависит данный класс. В противном случае,
/// конструкторы не смогут правильно работать с локализованными
/// ресурсами, сопоставленными данной форме.
/// </summary>
public ref class Form1 : public System::Windows::Forms::Form
{
public:
    Form1(void)
    {
        InitializeComponent();
        //
        //TODO: добавьте код конструктора
        //
    }

protected:
    /// <summary>
    /// Освободить все используемые ресурсы.
    /// </summary>
    ~Form1()
    {
        if (components)
        {
            delete components;
        }
    }

private: System::Windows::Forms::Button^ 3;
private: System::Windows::Forms::Button^  акрыть;
protected:

protected:

protected:

private: System::Windows::Forms::TextBox^ textBox1;
private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::Label^ label2;
private: System::Windows::Forms::TextBox^ textBox2;
private: System::Windows::Forms::Label^ label3;
private: System::Windows::Forms::TextBox^ textBox3;
private: System::Windows::Forms::Label^ label4;
private: System::Windows::Forms::TextBox^ textBox4;
private: System::Windows::Forms::Label^ label5;
private: System::Windows::Forms::TextBox^ textBox5;
private: System::Windows::Forms::Label^ label6;
private: System::Windows::Forms::TextBox^ textBox6;
private: System::Windows::Forms::Label^ label7;

```

```

private: System::Windows::Forms::TextBox^ textBox7;
private: System::Windows::Forms::Label^ label8;
private: System::Windows::Forms::TextBox^ textBox8;
private: System::Windows::Forms::Label^ label9;

private:
    /// <summary>
    /// Требуется переменная конструктора.
    /// </summary>
    System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
    /// <summary>
    /// Обязательный метод для поддержки конструктора - не изменяйте
    /// содержимое данного метода при помощи редактора кода.
    /// </summary>
    void InitializeComponent(void)
    {
        this->3 = (gcnew System::Windows::Forms::Button());
        this->акрыть = (gcnew System::Windows::Forms::Button());
        this->textBox1 = (gcnew System::Windows::Forms::TextBox());
        this->label1 = (gcnew System::Windows::Forms::Label());
        this->label2 = (gcnew System::Windows::Forms::Label());
        this->textBox2 = (gcnew System::Windows::Forms::TextBox());
        this->label3 = (gcnew System::Windows::Forms::Label());
        this->textBox3 = (gcnew System::Windows::Forms::TextBox());
        this->label4 = (gcnew System::Windows::Forms::Label());
        this->textBox4 = (gcnew System::Windows::Forms::TextBox());
        this->label5 = (gcnew System::Windows::Forms::Label());
        this->textBox5 = (gcnew System::Windows::Forms::TextBox());
        this->label6 = (gcnew System::Windows::Forms::Label());
        this->textBox6 = (gcnew System::Windows::Forms::TextBox());
        this->label7 = (gcnew System::Windows::Forms::Label());
        this->textBox7 = (gcnew System::Windows::Forms::TextBox());
        this->label8 = (gcnew System::Windows::Forms::Label());
        this->textBox8 = (gcnew System::Windows::Forms::TextBox());
        this->label9 = (gcnew System::Windows::Forms::Label());
        this->SuspendLayout();
        //
        // 3
        //
        this->3->Location = System::Drawing::Point(326, 12);
        this->3->Name = L"3";
        this->3->Size = System::Drawing::Size(75, 23);
        this->3->TabIndex = 0;
        this->3->Text = L"Решить";
        this->3->UseVisualStyleBackColor = true;
        this->3->Click += gcnew System::EventHandler(this,
&Form1::button1_Click);
        //
        // акрыть
        //

```



```

this->акрыть->Location = System::Drawing::Point(326, 54);
this->акрыть->Name = L"акрыть";
this->акрыть->Size = System::Drawing::Size(75, 22);
this->акрыть->TabIndex = 1;
this->акрыть->Text = L"Закрыть";
this->акрыть->UseVisualStyleBackColor = true;
this->акрыть->Click += gcnew System::EventHandler(this,
&Form1::button2_Click);
//
// textBox1
//
this->textBox1->Location = System::Drawing::Point(28, 217);
this->textBox1->Name = L"textBox1";
this->textBox1->Size = System::Drawing::Size(100, 20);
this->textBox1->TabIndex = 2;
this->textBox1->Text = L"Январь";
//
// label1
//
this->label1->AutoSize = true;
this->label1->Location = System::Drawing::Point(25, 190);
this->label1->Name = L"label1";
this->label1->Size = System::Drawing::Size(39, 13);
this->label1->TabIndex = 3;
this->label1->Text = L"месяц";
//
// label2
//
this->label2->AutoSize = true;
this->label2->Location = System::Drawing::Point(182, 190);
this->label2->Name = L"label2";
this->label2->Size = System::Drawing::Size(54, 13);
this->label2->TabIndex = 5;
this->label2->Text = L"действие";
//
// textBox2
//
this->textBox2->Location = System::Drawing::Point(185, 217);
this->textBox2->Name = L"textBox2";
this->textBox2->Size = System::Drawing::Size(100, 20);
this->textBox2->TabIndex = 4;
//
// label3
//
this->label3->AutoSize = true;
this->label3->Location = System::Drawing::Point(55, 17);
this->label3->Name = L"label3";
this->label3->Size = System::Drawing::Size(45, 13);
this->label3->TabIndex = 7;
this->label3->Text = L"х-целое";
//
// textBox3

```

```

//
this->textBox3->Location = System::Drawing::Point(28, 38);
this->textBox3->Name = L"textBox3";
this->textBox3->Size = System::Drawing::Size(100, 20);
this->textBox3->TabIndex = 6;
this->textBox3->Text = L"3";
//
// label4
//
this->label4->AutoSize = true;
this->label4->Location = System::Drawing::Point(193, 17);
this->label4->Name = L"label4";
this->label4->Size = System::Drawing::Size(89, 13);
this->label4->TabIndex = 9;
this->label4->Text = L"у-вещественное";
//
// textBox4
//
this->textBox4->Location = System::Drawing::Point(185, 38);
this->textBox4->Name = L"textBox4";
this->textBox4->Size = System::Drawing::Size(100, 20);
this->textBox4->TabIndex = 8;
this->textBox4->Text = L"1,53";
//
// label5
//
this->label5->AutoSize = true;
this->label5->Location = System::Drawing::Point(45, 68);
this->label5->Name = L"label5";
this->label5->Size = System::Drawing::Size(77, 13);
this->label5->TabIndex = 11;
this->label5->Text = L"через (double)";
//
// textBox5
//
this->textBox5->Location = System::Drawing::Point(28, 89);
this->textBox5->Name = L"textBox5";
this->textBox5->Size = System::Drawing::Size(100, 20);
this->textBox5->TabIndex = 10;
//
// label6
//
this->label6->AutoSize = true;
this->label6->Location = System::Drawing::Point(208, 68);
this->label6->Name = L"label6";
this->label6->Size = System::Drawing::Size(56, 13);
this->label6->TabIndex = 13;
this->label6->Text = L"через (int)";
//
// textBox6
//
this->textBox6->Location = System::Drawing::Point(185, 89);

```

```

this->textBox6->Name = L"textBox6";
this->textBox6->Size = System::Drawing::Size(100, 20);
this->textBox6->TabIndex = 12;
//
// label7
//
this->label7->AutoSize = true;
this->label7->Location = System::Drawing::Point(168, 124);
this->label7->Name = L"label7";
this->label7->Size = System::Drawing::Size(125, 13);
this->label7->TabIndex = 17;
this->label7->Text = L"чекер Convert::ToInt16()";
//
// textBox7
//
this->textBox7->Location = System::Drawing::Point(185, 145);
this->textBox7->Name = L"textBox7";
this->textBox7->Size = System::Drawing::Size(100, 20);
this->textBox7->TabIndex = 16;
//
// label8
//
this->label8->AutoSize = true;
this->label8->Location = System::Drawing::Point(55, 124);
this->label8->Name = L"label8";
this->label8->Size = System::Drawing::Size(0, 13);
this->label8->TabIndex = 15;
//
// textBox8
//
this->textBox8->Location = System::Drawing::Point(28, 145);
this->textBox8->Name = L"textBox8";
this->textBox8->Size = System::Drawing::Size(100, 20);
this->textBox8->TabIndex = 14;
//
// label9
//
this->label9->AutoSize = true;
this->label9->Location = System::Drawing::Point(12, 124);
this->label9->Name = L"label9";
this->label9->Size = System::Drawing::Size(135, 13);
this->label9->TabIndex = 18;
this->label9->Text = L"чекер Convert::ToDouble()";
//
// Form1
//
this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
this->AutoScaleMode
System::Windows::Forms::AutoScaleMode::Font;
this->ClientSize = System::Drawing::Size(413, 266);
this->Controls->Add(this->label9);
this->Controls->Add(this->label7);

```

=

```

        this->Controls->Add(this->textBox7);
        this->Controls->Add(this->label8);
        this->Controls->Add(this->textBox8);
        this->Controls->Add(this->label6);
        this->Controls->Add(this->textBox6);
        this->Controls->Add(this->label5);
        this->Controls->Add(this->textBox5);
        this->Controls->Add(this->label4);
        this->Controls->Add(this->textBox4);
        this->Controls->Add(this->label3);
        this->Controls->Add(this->textBox3);
        this->Controls->Add(this->label2);
        this->Controls->Add(this->textBox2);
        this->Controls->Add(this->label1);
        this->Controls->Add(this->textBox1);
        this->Controls->Add(this->акрыць);
        this->Controls->Add(this->3);
        this->Name = L"Form1";
        this->Text = L"Form1";
        this->Load += gcnw System::EventHandler(this,
&Form1::Form1_Load);
        this->ResumeLayout(false);
        this->PerformLayout();

    }
#pragma endregion
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^
e) {

        int x = Convert::ToInt16(textBox3->Text);
        textBox5->Text = Convert::ToString((double)x);
        textBox8->Text = Convert::ToString(Convert::ToDouble(x));

        double y = Convert::ToDouble(textBox4->Text);
        textBox6->Text = Convert::ToString((int)y);
        textBox7->Text = Convert::ToString(Convert::ToInt16(y));

        if (textBox1->Text != "")
    { month z; String^ h="";
        h=textBox1->Text;
        if (h=="Январь") z=Январь;
        if (h=="Июнь") z=Июнь;
        if (h=="Сентябрь") z=Сентябрь;

            if (z == Январь)
    {
        textBox2->Text = "Новый год!";
    }
            else if ((z == Июнь) | (z == Июль) | (z == Август))
    {
        textBox2->Text = "Едем на море!";
    }
    }
    }

```

```

    }
    else
    {
        textBox2->Text = "Учимся :(";
    }
}
else
{
    textBox2->Text = "Ошибка, поле месяц - пусто!";
}
}
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {
    Form1::Close();
}
private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e)
{
}
};
}

```

### Визуализация работы программы.

Рис. 2. Визуализация работы программы совместимость типов.

### Задача3. Вычисление функции $\exp(x)$ .

Вычисление ряда  $1 + x + x^2/2! + x^3/3! + \dots$  для заданного аргумента  $X$  с заданной точностью  $\epsilon$ . Результат напечатать с 3 знаками после запятой.

Исходные данные: 1.0 0.001

Результат: 2.718

### Код программы на языке Си# (приложение Windows Forms).

```

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 WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            double a = 1, s = 0;
            double eps = Convert.ToDouble(textBox1.Text);
            double x = Convert.ToDouble(textBox2.Text); string y = " ";
            int i = 1;
            while((a*x/1)/(s+ a)> eps)
            {
                s = s + a;
                a = a * x / i;
                i++;
            }
            y = Convert.ToString(s);
            textBox3.Text = y;
        }
        private void button2_Click(object sender, EventArgs e)
        {
            Form1.ActiveForm.Close();
        }
    }
}

```

### Визуализация работы программы.

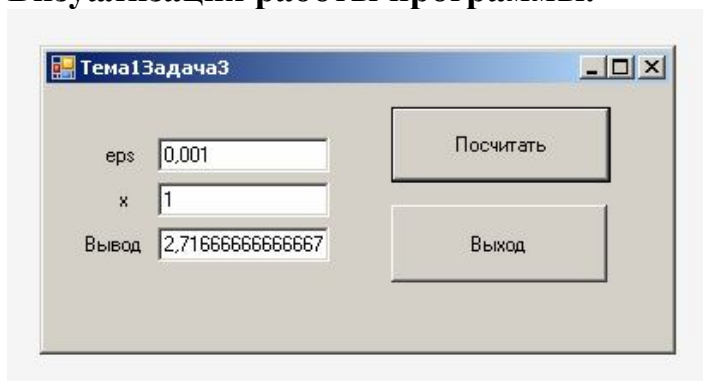


Рис. 3. Визуализация работы программы вычисление exp.

### Задача4. Квадратное уравнение.

Даны вещественные коэффициенты квадратного уравнения  $A \cdot X^2 + B \cdot X + C = 0$ . Коэффициент  $A \neq 0$ . Написать программу для вычисления и печати корней этого уравнения. Ответ напечатать с точностью 2 знака после запятой. Если корней не существует, напечатать слово «НЕТ».

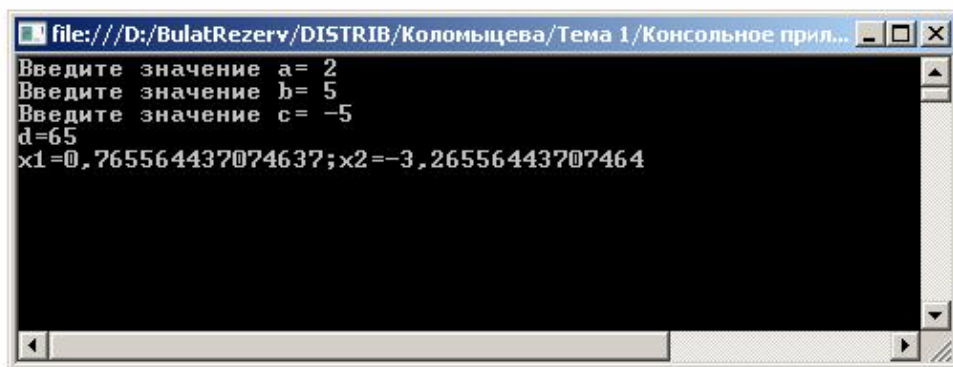
### Код программы на языке Си# (консольное приложение).

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Введите значение a = ");
            int a =
                Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Введите значение b = ");
            int b =
                Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Введите значение c = ");
            int c =
                Convert.ToInt32(Console.ReadLine());
            double d;
            d = b * b - 4 * a * c;
            Console.WriteLine("d=" + d);
            if (d < 0)

                {
                    Console.WriteLine("Решение невозможно, дискриминант отрицателен.");
                }
            if (d == 0)
                {
                    double x1 = -b / (2 * a);
                    Console.WriteLine("x1= "+x1 );
                }
            if (d > 0)
                {
                    double x1 = (-b + Math.Sqrt(Convert.ToDouble(d))) / (2 * a);
                    double x2 = (-b - Math.Sqrt(Convert.ToDouble(d))) / (2 * a);
                    Console.WriteLine("x1=" + x1 + ";x2=" + x2);
                }
            Console.ReadKey();
        }
    }
}
```

### Визуализация работы программы.



```
file:///D:/BulatRezerv/DISTRIB/Коломыцева/Тема 1/Консольное прил...
Введите значение a= 2
Введите значение b= 5
Введите значение c= -5
d=65
x1=0,765564437074637;x2=-3,26556443707464
```

Рис. 4. Визуализация работы программы решение уравнения.

## ТЕМА 2: ГЕОМЕТРИЯ

### Цель.

Научиться применять знания по геометрии при решении задач с использованием компьютеров (программ). Уметь выполнять приближённые вычисления с точностью. Уметь рисовать кривые на форме.

### Основные понятия.

Вычисления с вещественными числами с заданной точностью. Использование математических функций:  $\sin$ ,  $\cos$ ,  $\text{fabs}$ ,  $\exp$ ,  $\log$

### Ключевые слова.

Точка, уравнение прямой линии, геометрическая фигура.

### Задача1. Прямая и точка.

Составить уравнение прямой  $AX + BY + C = 0$ , проходящей через 2 заданные точки  $(X_1, Y_1)$  и  $(X_2, Y_2)$ . Напечатать полученные коэффициенты этой прямой. Далее взять некую точку  $(X_3, Y_3)$  и найти кратчайшее расстояние от точки до прямой. Нарисовать на форме полученные прямую и точку.

### Код программы на языке Си#.

```
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 Прямая
{
    public partial class Form1 : Form
    {
```



```

public Form1()
{
    InitializeComponent();
}
private void button1_Click(object sender, EventArgs e)
{
    double x1 = Convert.ToDouble(textBox1.Text);
    double x2 = Convert.ToDouble(textBox2.Text);
    double x3 = Convert.ToDouble(textBox3.Text);
    double y3= Convert.ToDouble(textBox6.Text);
    double y1 = Convert.ToDouble(textBox4.Text);
    double y2 = Convert.ToDouble(textBox5.Text);
    double A = y2-y1;
    double B = x1-x2;
    double C = (y1 - y2)*x1+(x2- x1)*y1;
    double k = Math.Abs (A*x3+B*y3+ C) /((Math.Sqrt (Math.Pow(A, 2) +
Math.Pow(B, 2)));
    textBox10.Text = Convert.ToString(Math.Abs(k));
    textBox7.Text = Convert.ToString(A);
    textBox8.Text = Convert.ToString(B);
    textBox9.Text = Convert.ToString(C);
    Graphics de=this.CreateGraphics();
    int x11 = Convert.ToInt32(textBox1.Text);
    int x21 = Convert.ToInt32 (textBox2.Text);
    int x31 = Convert.ToInt32 (textBox3.Text);
    int y31 = Convert.ToInt32(textBox6.Text);
    int y11 = Convert.ToInt32(textBox4.Text);
    int y21= Convert.ToInt32(textBox5.Text);
    Pen p = new Pen(Color.IndianRed, 8);
    Point p1 = new Point(x11, y11);
    Point p2 = new Point(x21, y21);
    Point p3 = new Point(x31, y31);
    Point p4 = new Point(x31 + 4, y31 + 4);
    de.DrawLine(p, p1, p2);
    de.DrawLine(p, p3, p4);
}
private void button2_Click(object sender, EventArgs e)
{
    Form1.ActiveForm.Close();
}
private void Form1_Paint(object sender, PaintEventArgs e)
{
    /*Graphics gr =e.Graphics;
    Pen p = new Pen(Color.IndianRed, 8);
    Point p1 = new Point(400, 600);
    Point p2 = new Point(800, 600);
    Point p3 = new Point(700, 300);
    gr.DrawLine(p, p1, p2);*/
}
private void button3_Click(object sender, EventArgs e)
{

```

```

Graphics de = this.CreateGraphics();
int x11 = Convert.ToInt32(textBox1.Text);
int x21 = Convert.ToInt32(textBox2.Text);
int x31 = Convert.ToInt32(textBox3.Text);
int y11 = Convert.ToInt32(textBox6.Text);
int y11 = Convert.ToInt32(textBox4.Text);
int y21 = Convert.ToInt32(textBox5.Text);
Pen p = new Pen(Color.White, 8);
Point p1 = new Point(x11, y11);
Point p2 = new Point(x21, y21);
Point p3 = new Point(x31, y31);
Point p4 = new Point(x31 + 4, y31 + 4);
de.DrawLine(p, p1, p2);
de.DrawLine(p, p3, p4);
}
}
}

```

## Визуализация работы программы.

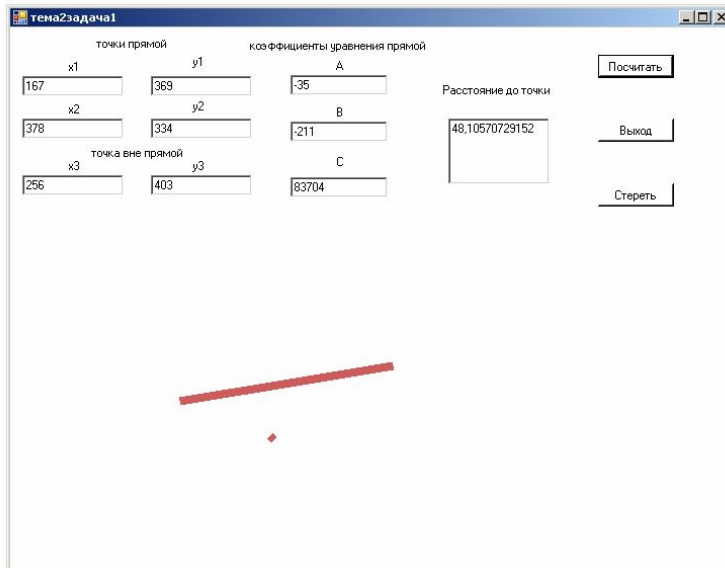


Рис. 5. Визуализация работы программы прямая и точка.

## Задача2. Треугольник.

На плоскости три точки заданы своими целыми (для простоты) координатами. Они образуют невырожденный треугольник. Вычислить площадь и периметр этого треугольника и нарисовать его на форме.

### Алгоритм.

Можно воспользоваться одной из формул, знакомых со школы. Для этого придётся вычислить какие-то элементы треугольника – стороны, углы, периметр. Мы воспользуемся формулой, дающей значение площади со знаком в зависимости от направления обхода вершин треугольника – через определитель 3-го порядка.

$$S = \frac{|1 \ x_1 \ y_1|}{|1 \ x_2 \ y_2|} * 0.5$$

$$|1 \ x_3 \ y_3|$$

### Код программы на языке Си++.

```
#pragma once
.....
#pragma endregion
    int x1,x2,x3,y1,y2,y3;
    private:      System::Void      Form1_Load(System::Object^      sender,
System::EventArgs^ e)
        {
        }
    private:      System::Void      button1_Click(System::Object^      sender,
System::EventArgs^ e)
        {
            x1=System::Int16::Parse(textBox1->Text);
            y1=System::Int16::Parse(textBox2->Text);
            x2=System::Int16::Parse(textBox3->Text);
            y2=System::Int16::Parse(textBox4->Text);
            x3=System::Int16::Parse(textBox5->Text);
            y3=System::Int16::Parse(textBox6->Text);

            double L1 = Math::Sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));
            double L2 = Math::Sqrt((x3-x2)*(x3-x2)+(y3-y2)*(y3-y2));
            double L3 = Math::Sqrt((x1-x3)*(x1-x3)+(y1-y3)*(y1-y3));

            double P = L1+L2+L3;
            double p = P/2;

            P=Math::Round(P,2);
            textBox7->Text = P.ToString();

            double S = Math::Sqrt(p*(p-L1)*(p-L2)*(p-L3));
            S=Math::Round(S,2);
            textBox8->Text = S.ToString();
        }
    private:      System::Void      button2_Click(System::Object^      sender,
System::EventArgs^ e)
        {
            Form1::Close();
        }
    private:      System::Void      Form1_Paint(System::Object^      sender,
System::Windows::Forms::PaintEventArgs^ e)
        {

        }

private: System::Void button3_Click(System::Object^ sender, System::EventArgs^
e)
    {
```

```

x1=System::Int16::Parse(textBox1->Text);
y1=System::Int16::Parse(textBox2->Text);
x2=System::Int16::Parse(textBox3->Text);
y2=System::Int16::Parse(textBox4->Text);
x3=System::Int16::Parse(textBox5->Text);
y3=System::Int16::Parse(textBox6->Text);

Graphics^g1=this->CreateGraphics();
g1->Clear(System::Drawing::Color::WhiteSmoke);
g1->DrawLine(System::Drawing::Pens::Green,x1,y1,x2,y2);
g1->DrawLine(System::Drawing::Pens::Green,x2,y2,x3,y3);
g1->DrawLine(System::Drawing::Pens::Green,x3,y3,x1,y1);
g1->DrawEllipse(System::Drawing::Pens::Black,x1,y1,1,1);
g1->DrawEllipse(System::Drawing::Pens::Black,x2,y2,1,1);
g1->DrawEllipse(System::Drawing::Pens::Black,x3,y3,1,1);
}
};
}

```

### Визуализация работы программы.

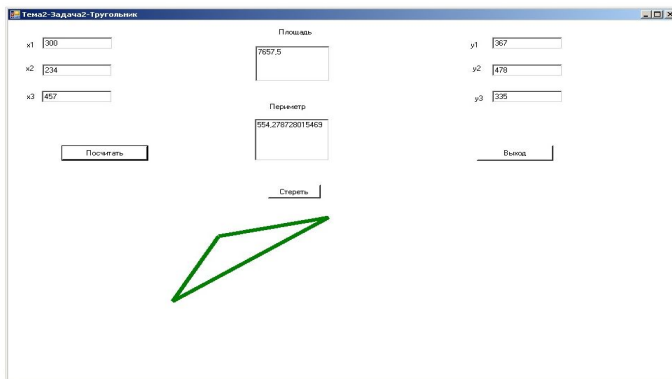


Рис. 6. Визуализация работы программы треугольник.

### Задача3. Четырехугольник.

Даны координаты четырёх точек  $(X_1, Y_1)$ ,  $(X_2, Y_2)$ ,  $(X_3, Y_3)$ ,  $(X_4, Y_4)$  в порядке обхода по часовой стрелке. Определить и напечатать тип (название) этого четырёхугольника – ромб, квадрат, параллелограмм, трапеция, прямоугольник или просто четырехугольник. Нарисовать четырехугольник на форме.

### Код программы на языке Си#.

```

#pragma once
.....
#pragma endregion
private: System::Void Form1_Load(System::Object^ sender,
System::EventArgs^ e)
{
}
private: System::Void button2_Click(System::Object^ sender,
System::EventArgs^ e)
{
}

```

```

        Form1::Close();
    }
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^
e)
    {
    int x1,x2,x3,x4,y1,y2,y3,y4;
        x1=System::Int32::Parse(textBox1->Text);
    y1=System::Int32::Parse(textBox2->Text);
    x2=System::Int32::Parse(textBox3->Text);
    y2=System::Int32::Parse(textBox4->Text);
    x3=System::Int32::Parse(textBox5->Text);
    y3=System::Int32::Parse(textBox6->Text);
    x4=System::Int32::Parse(textBox7->Text);
        y4=System::Int32::Parse(textBox8->Text);

    int L1,L2,L3,L4;
    L1=Math::Sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));
    L2=Math::Sqrt((x3-x2)*(x3-x2)+(y3-y2)*(y3-y2));
    L3=Math::Sqrt((x4-x3)*(x4-x3)+(y4-y3)*(y4-y3));
    L4=Math::Sqrt((x4-x1)*(x4-x1)+(y4-y1)*(y4-y1));

    int D1,D2;
    D1=Math::Sqrt((x2-x4)*(x2-x4)+(y2-y4)*(y2-y4));
    D2=Math::Sqrt((x1-x3)*(x1-x3)+(y1-y3)*(y1-y3));

    Graphics^g = this->CreateGraphics();
    g->Clear(System::Drawing::Color::WhiteSmoke);
    g->DrawLine(System::Drawing::Pens::Green,x1,y1,x2,y2);
    g->DrawLine(System::Drawing::Pens::Green,x2,y2,x3,y3);
    g->DrawLine(System::Drawing::Pens::Green,x3,y3,x4,y4);
    g->DrawLine(System::Drawing::Pens::Green,x1,y1,x4,y4);

    int f=0;
    if(((L1==L2)&&(L2==L3)&&(L3==L4)&&(L1==L4))&&(D1==D2) )
    {
        f=1;
    }else
    {
        if((D1==D2)&&(L1==L3&&L2==L4))
        {
            f=2;
        }else
        {
            if((L1==L2)&&(L2==L3)&&(L3==L4)&&(L1==L4))
            {
                f=3;
            }else
            {
                if(L1==L3&&L2==L4)
                {

```



Даны два одинаковых круга. В клетках окна содержатся координаты  $(x_1, y_1)$  и  $(x_2, y_2)$  - центры двух кругов и задан общий радиус  $r$ . Все числа целые и удовлетворяют следующим ограничениям:  $1 \leq x_1, y_1, x_2, y_2, r \leq 100$ . Найти суммарную площадь этих кругов с точностью  $-0.001$ . Нарисовать окружности на форме.

### Код программы на языке Си++.

```
#pragma once
.....
#pragma endregion
int x1,x2,y1,y2,R;
double S;
private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e)
{
    x1=System::Int16::Parse(textBox1->Text);
    y1=System::Int16::Parse(textBox2->Text);
    x2=System::Int16::Parse(textBox3->Text);
    y2=System::Int16::Parse(textBox4->Text);
    R=System::Int16::Parse(textBox5->Text);
    double d = Math::Sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));
    double a = Math::Acos((d/2)/R);
    double s = R*R*(2*a-Math::Sin(2*a));
    if (d < (R*2))
    {
        S=2*R*R*Math::PI-s;
    }else
    {
        S=2*R*R*Math::PI;
    }
    S=Math::Round(S,3);
    textBox6->Text=S.ToString();
}

private: System::Void button3_Click(System::Object^ sender, System::EventArgs^
e)
{
    x1=System::Int16::Parse(textBox1->Text);
    y1=System::Int16::Parse(textBox2->Text);
    x2=System::Int16::Parse(textBox3->Text);
    y2=System::Int16::Parse(textBox4->Text);
    R=System::Int16::Parse(textBox5->Text);

    Graphics^g = this->CreateGraphics();
    g->Clear(System::Drawing::Color::WhiteSmoke);
    g->DrawEllipse(System::Drawing::Pens::Black,x1,y1,2*R,2*R);
    g->DrawEllipse(System::Drawing::Pens::Black,x2,y2,2*R,2*R);
```

```

    }
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {
    Form1::Close();
}
};
}

```

### Визуализация работы программы.

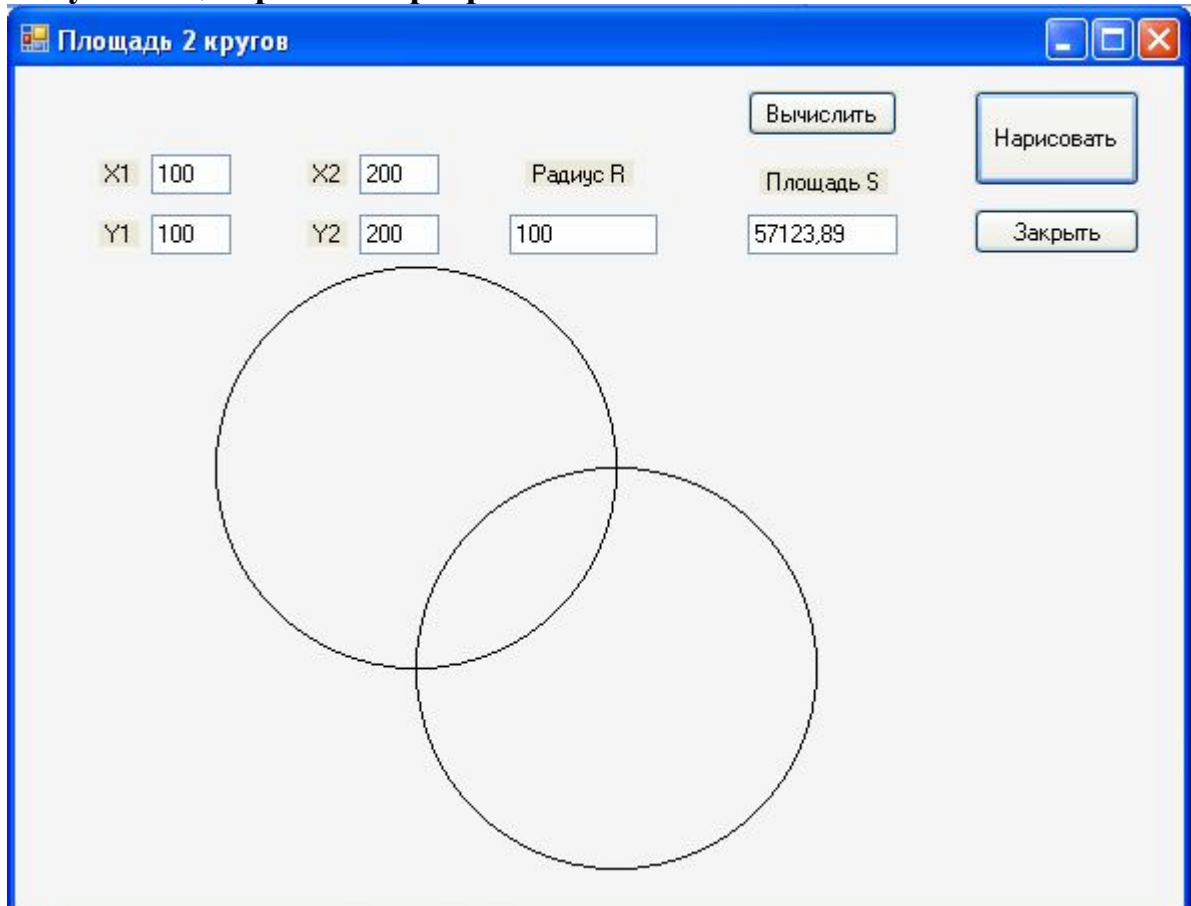


Рис. 8. Визуализация работы программы окружность.

## ТЕМА 3: ПОСЛЕДОВАТЕЛЬНОСТИ

### Цель.

Умение обрабатывать большие последовательности данных (значений) по одному без одновременного их хранения в памяти.

### Основные понятия.

Многократное использование переменных. Однократная обработка данных.

### Ключевые слова.

Последовательность, цикл.

### Задача1. Поиск минимума и максимума.



Дана последовательность целых чисел. Найти самое большое, самое маленькое число и сумму всех чисел. Далее отсортировать последовательность в порядке возрастания (сортировка пузырьком). В начале последовательности задаётся количество чисел в ней.

**Алгоритм.**

Будем вводить по одному числу и сравнивать его с эталонами - наибольшим и наименьшим. Сначала в качестве обоих эталонов возьмём первое число. Все числа хранить в памяти не будем, каждое новое будет помещаться в ту же ячейку, где хранилось предыдущее.

**Код программы на языке Си++.**

```
#pragma once
#include<String>
.....
#pragma endregion
private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e) {
    }
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {
    Form1::Close();
}
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    const int N=1000;
    int a[N];
    array<wchar_t>^mas=textBox1->Text->ToCharArray();
    int l = 0;
    String^h="";String^h1;
    int SUM = 0;
    int n = mas->Length;
    int j = 0;
    for(int i = 0;i<n;i++)
    {
        /*          h1 =h1 + Convert::ToString(mas[i+i]);
                   a[i] = Convert::ToInt16(h1);
                   h = h + Convert::ToString(a[i]+" ");
        */
        if(Convert::ToString(mas[i])!=" ")
        {
            h1 =Convert::ToString(mas[i]);
            l = l*10 + Convert::ToInt16(h1);
            if(i==n-1)
            {
                a[j] = l;
                SUM+=a[j];
            }
            }else
            {
                a[j] = l;
                SUM+=a[j];
            }
        }
    }
}
```

```

        l=0;
        j++;
    }
}
textBox2->Text = Convert::ToString(j+1);
int MAX = a[0];
int MIN=a[0];
for(int i = 0;i<=j;i++)
{
    if(MAX <= a[i])
    {
        MAX = a[i];
    }
    if(MIN > a[i])
    {
        MIN = a[i];
    }
}
textBox3->Text = Convert::ToString(MAX);
textBox4->Text = Convert::ToString(MIN);
textBox5->Text = Convert::ToString(SUM);
for(int i = 0;i<=j;i++)
{
    for(int k = i+1;k<=j;k++)
    {
        if(a[i] > a[k])
        {
            int temp = a[i];
            a[i] = a[k];
            a[k] = temp;
        }
    }
}

for(int i = 0;i<=j;i++)
    h += Convert::ToString(a[i])+" ";
textBox6->Text = h;
}
};
}

```

### **Задача2. Поиск пиков.**

В последовательности целых чисел найти номера и значения всех элементов, которые строго больше предыдущего и последующего элементов. Первое и последнее число не учитываются, т.к. у них только по одному соседу. В окне задаётся количество чисел в последовательности и сами числа.

### Визуализация работы программы.

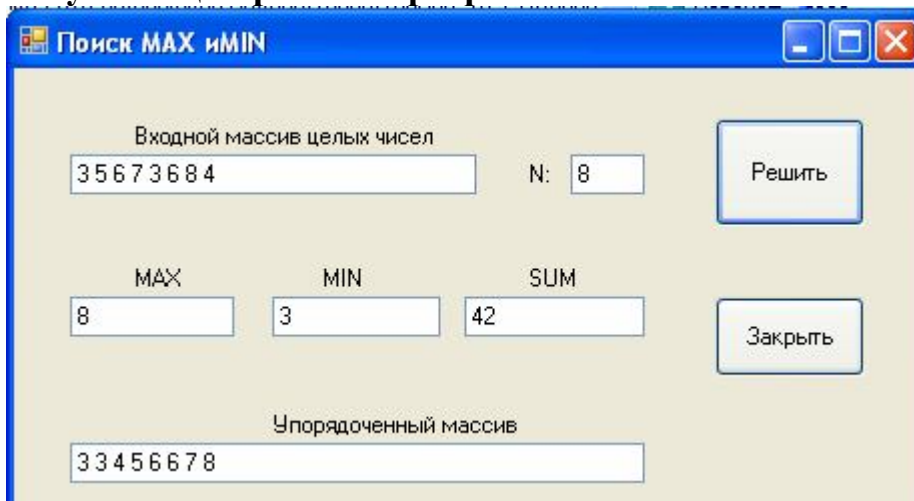


Рис. 9. Визуализация работы программы поиск мин макс.

### Код программы поиск пиков на языке Си++.

### Визуализация работы программы.

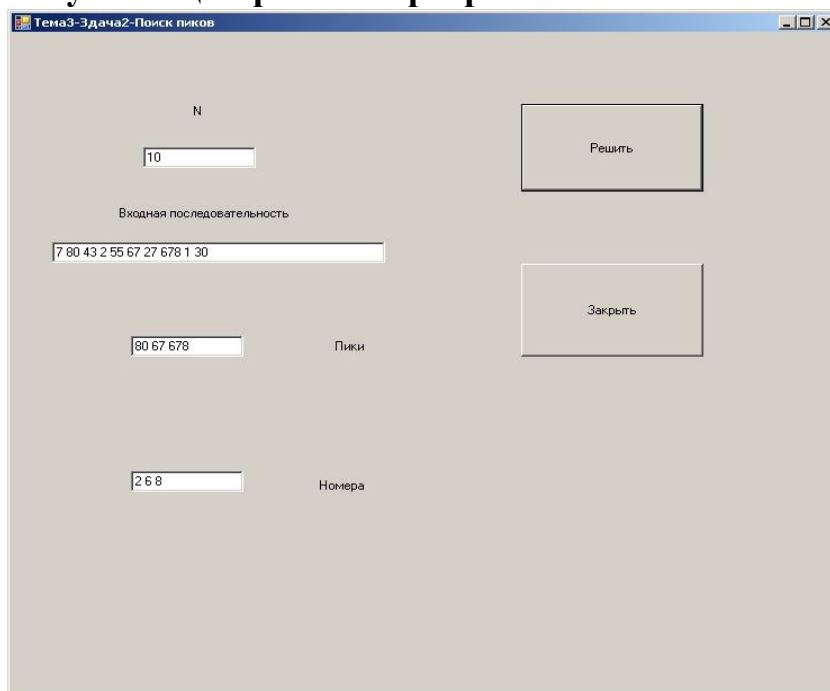


Рис. 10. Визуализация работы программы поиск пиков.

### Задача3. Обработка двоичной последовательности.

Дано натуральное число N. Перевести это число в двоичный вид и найти в полученной последовательности из 0 и 1 самую длинную непрерывную цепочку нулей и единиц. В единственной строке записана последовательность нулей и единиц (без пробелов). Суммарное количество цифр не превышает 1000.

Вывести искомую длину цепочки нулей.

### Код программы на языке Си++.

## Визуализация работы программы.

Тема3-Задача3-Двоичная-последовательность

N

30

Передаваемое число

300

Передаваемая последовательность

100101100

длина 2 "0" номера 28 "0"

длина 2 "1" номера 6 "1"

Посчитать

Закреть

Рис. 11. Визуализация работы программы двоичная последовательность.

### Задача 4. Минимальная вместимость автобуса.

На автобусном маршруте из пункта А в Б имеется  $n$  остановок. На каждой остановке из автобуса сначала выходят  $a_i$  человек, потом входят  $b_i$  человек. Изначально автобус пустой. На обратном пути из Б в А сначала выходят  $c_i$  человек, потом входят  $d_i$  человек. Определить и напечатать минимальную возможную вместимость автобуса.

### Код программы на языке Си#.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace VanAnd_T3_Z4
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
        }
        private void button2_Click(object sender, EventArgs e)
        {
            Form1.ActiveForm.Close();
        }
        private void button1_Click(object sender, EventArgs e)
        {
        }
    }
}
```

```

{
    int K = 0; //Вместимость
    int sum = 0;
    string[] sNum1 = textBox1.Text.Split();
    string[] sNum2 = textBox2.Text.Split();
    string[] sNum3 = textBox3.Text.Split();
    string[] sNum4 = textBox4.Text.Split();
    int N = sNum1.Length;
    textBox6.Text = Convert.ToString(N);
    int[] a = new int[N];
    int[] b = new int[N];
    for (int i = 0; i < N; i++)
    {
        a[i] = int.Parse(sNum1[i]);
        b[i] = int.Parse(sNum2[i]);
    }

    K = a[0]; sum = b[0];

    for (int i = 0; i < N; i++)
    {
        sum -= b[i];
        sum += a[i];
        if (K < sum)
            K = sum;
    }
    for (int i = N-1; i > 0; i--)
    {
        a[N-i] = int.Parse(sNum3[i]);
        b[N-i] = int.Parse(sNum4[i]);
    }
    for (int i = 1; i < N; i++)
    {
        sum -= b[i];
        sum += a[i];
        if (K < sum)
            K = sum;
    }
    textBox5.Text = Convert.ToString(K);
}
}
}

```

## Визуализация работы программы.

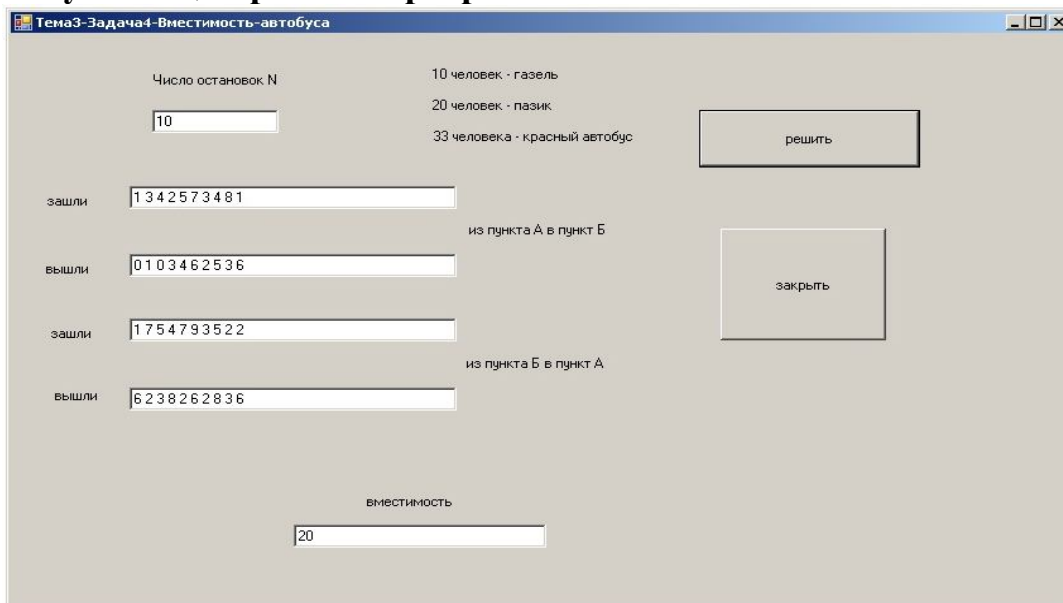


Рис. 12. Визуализация работы программы автобус.

## ТЕМА 4: ОДНОМЕРНЫЕ МАССИВЫ

### Цель.

Изучение алгоритмов обработки большого количества данных, хранящихся одновременно в оперативной памяти.

### Основные понятия.

Массив – упорядоченное множество однотипных элементов, занимающих смежные участки оперативной памяти. Они используются в тех случаях, когда элементы последовательности должны обрабатываться не в том порядке, в котором поступают в программу, или каждый элемент должен обрабатываться (просматриваться) несколько раз. Имя массива ассоциируется с адресом начала массива (первого элемента). Этот адрес является постоянным. Основная операция над массивами – операция доступа к элементу []. Можно использовать значение элемента и можно его изменять. Используя адреса можно обращаться к элементам массива с помощью операции \* и операции адресной арифметики.

Ввод и вывод массивов осуществляется поэлементно (для символьных строк – см. ниже).

### Ключевые слова.

Массив, индекс, элемент, счётчик, адрес, указатель, случайное число.

### Задача1. Сжатие массива.

Сжать неупорядоченный массив (удалить повторяющиеся значения). В конец поместить нули. В первой строке задаётся размер массива, во второй – сам массив.

## Код программы на языке Си++.

```
#pragma once
.....
#pragma endregion
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e)
{
    Form1::Close();
}
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e)
{
    const int N=1000;
    int a[N];

    array<wchar_t>^mas=textBox1->Text->ToCharArray();

    int l = 0;
    String^h="";String^h1;
    int n = mas->Length;
    int j = 0;
    for(int i = 0;i<n;i++)
    {
        if(Convert::ToString(mas[i])!=" ")
        {
            h1 =Convert::ToString(mas[i]);
            l = l*10 + Convert::ToInt16(h1);
            if(i==n-1)
            {
                a[j] = l;
            }
            }else
            {
                a[j] = l;
                l=0;
            }
            j++;
        }
    }
    int count0 = j+1;
    for(int i = 0;i<j;)
    {
        if(a[i] == a[i+1])
        {
            for(int k = i+1;k<j;k++)
            {
                a[k] = a[k+1];
            }
            j--;
        }else
            i++;
    }
    for(int i = 0;i <= j;i++)
```

```

    {
        h+=Convert::ToString(a[i])+" ";
    }

    textBox3->Text = Convert::ToString(100 - (((j+1)*100)/count0))+ "%";
    for(int i = j+1;i<count0;i++)
        h+=Convert::ToString(0)+" ";
    textBox2->Text = h ;
}
private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e)
{
}
};
}

```

### Визуализация работы программы.

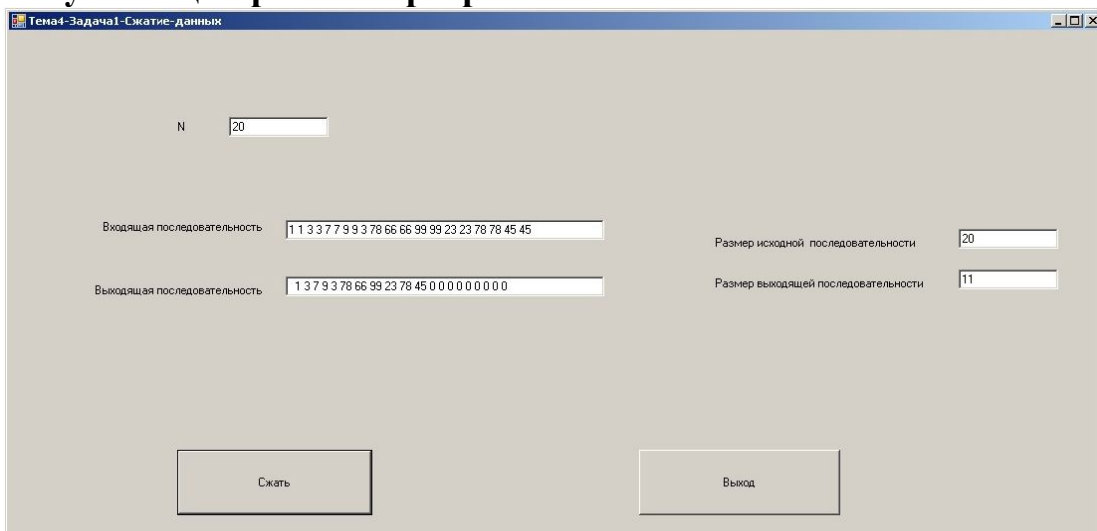


Рис. 13. Визуализация работы программы сжатие массива.

### Задача2. Количество цифр в числах.

Рассматриваются все числа от 1 до n. Сколько раз встречается каждая цифра от 0 до 9 во всех этих числах вместе взятых? Вводится натуральное n. Нужно вывести 10 чисел в строке: количество нулей, единиц и т.д. девяток.

#### Код программы на языке Си++.

### Задача3. Случайные числа.

Сгенерировать случайный набор целых и вещественных чисел  $X_1, X_2, \dots, X_n$ . Для набора вещественных чисел найти целое значение A такое, что сумма  $(|X_1-A| + |X_2-A| + \dots + |X_n-A|)$  минимальна, т.е. такую точку A, до которой сумма расстояний от заданных точек минимальна. В наборе целых чисел для заданных номеров k1 и k2 ( $1 \leq k_1 \leq k_2 \leq N$ ) переставить элементы массива от k1-го до k2-го в обратном порядке.

#### Алгоритм.

Для набора вещественных чисел алгоритм уже дан. Для набора целых чисел в цикле запустим два счётчика навстречу друг другу и соответствующие



элементы будем менять местами.

### **Код программы на языке Си++( консольное приложение Win32).**

```
#include "stdafx.h"
#include "iostream"
#include <ctime>

using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    const int N = 1000;
    double a[N];
    int b[N];
    int n;
    double A = 0;
    int s = 0;
    int K1,K2;
    int min = 0;
    srand(time(0));
    cout<<"Vvedite n\n";
    cin>>n;

    cout<<"Vvedite "<<n<<" veshestvennih chisla\n";
    for(int i = 0;i<n;i++)
    {
        cin>>a[i];
        s+=a[i];
    }
    s=s/3;
    double ss=abs(a[0]-s);

    for(int i =0;i<3;i++)
    {
        if(abs(a[i]-s)<=ss)
        {
            ss=abs(a[i]-s);
            A = a[i];
        }
    }

    for(int i = 0;i<3;i++)
    {
        min += abs(a[i]-A);
    }

    cout<<"\nA="<<A<<" min="<<min<<endl;
    s=0;
    for(int i = 0;i<n;i++)
    {
```

```

        b[i] = rand()%100;
        a[i] = 0.01*(rand()%10000);
        s+=b[i];
    }
    cout<<"\nMassiv celih chisel:"<<"\n";
    for(int i = 0;i<n;i++)
        cout<<b[i]<<" ";

    cout<<"\n\n"<<"Massiv veshestvennih chisel:\n";

    for(int i = 0;i<n;i++)
        cout<<a[i]<<" ";

    s = s/n;
    A = 0;
    ss =abs(a[0]-s);
    min = 0;

    for(int i =0;i<n;i++)
    {
        if(abs(a[i]-s)<=ss)
        {
            ss=abs(a[i]-s);
            A = a[i];
        }
    }

    for(int i = 0;i<n;i++)
    {
        min +=abs(a[i]-A);
    }

    cout<<"\n\nChislo A = "<<A<<"\n"<<"Minimum="<<min<<" s="<<s;
    cout<<"\nVvedite K1 "<<"\n";
    cin>>K1;
    cout<<"Vvedite K2 "<<"\n";
    cin>>K2;

    for(int i = K1-1;i<K2; i++,K2--)
    {
        double temp=b[i];
        b[i] = b[K2-1];
        b[K2-1] = temp;
    }

    for(int i = 0;i<n;i++)
        cout<<b[i]<<" ";

    cin>>s;

    return 0;
}

```

## Визуализация работы программы количество цифр.

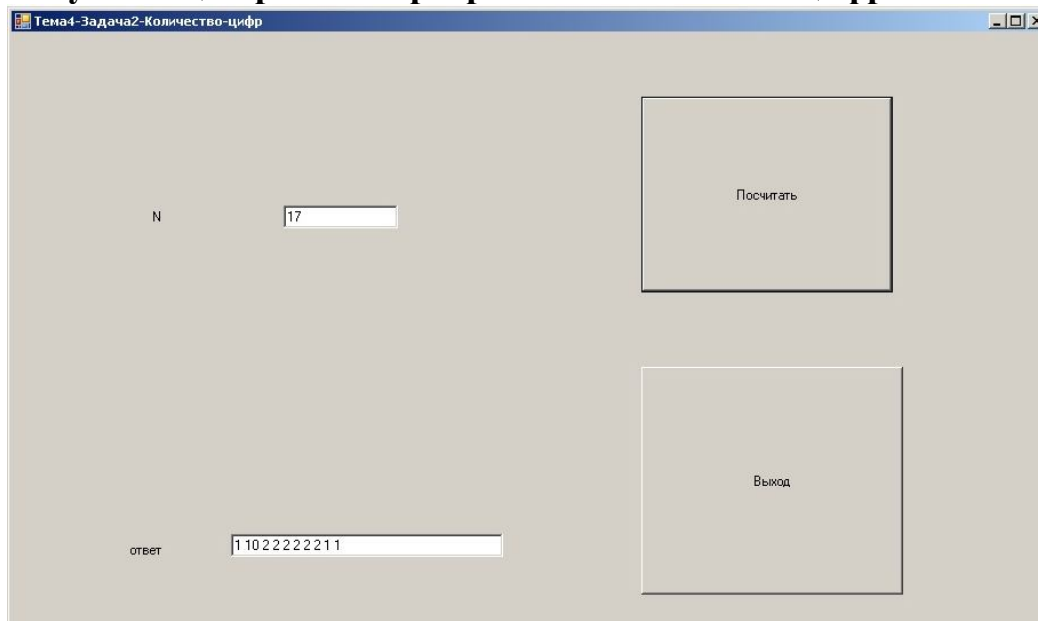


Рис. 14. Визуализация работы программы количество цифр.

## Визуализация работы программы случайные числа.

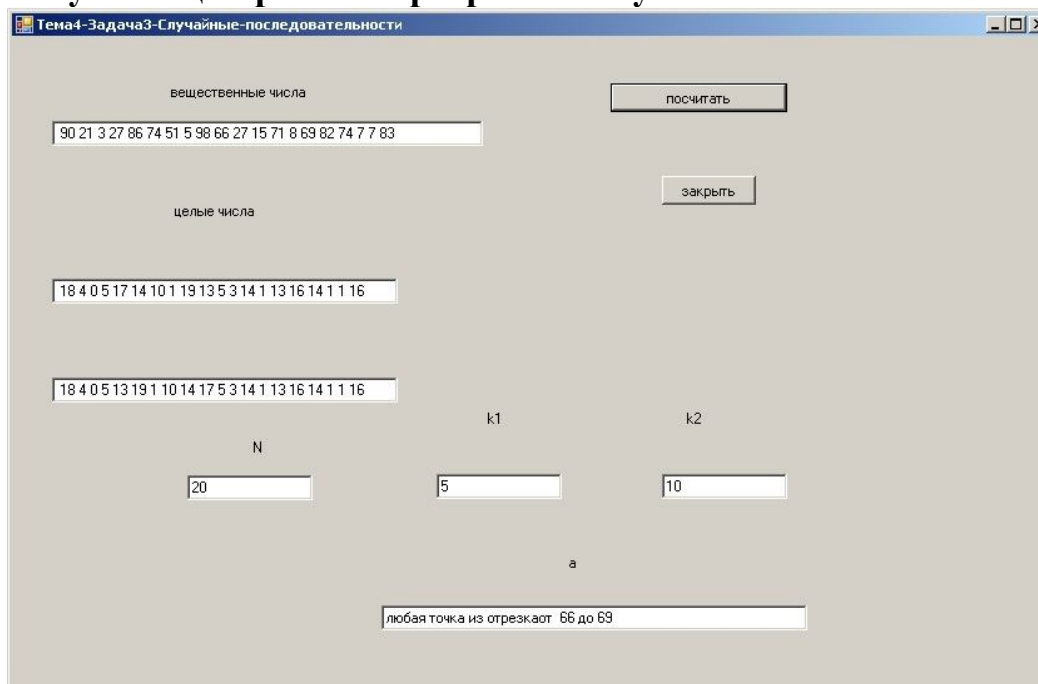


Рис. 15. Визуализация работы программы случайные числа.

### Задача4. Секретное сообщение.

На секретную базу в Арктике поступила шифровка – последовательность из  $n+3$  десятичных цифр. Она содержит код целостности полученной шифровки, код идентификации и полезную информацию.

В полезной информации может содержаться номер секретной базы в Антарктиде, который является последовательностью из  $k$  десятичных цифр. При этом для того, чтобы отличить его от ненужной Вам информации, он повторен в шифровке хотя бы два раза (возможно, эти два вхождения

перекрываются).

Написать программу, которая определяет целостность полученной шифровки, идентифицирует ее, и по шифровке и длине номера секретной базы определяет, содержит ли шифровка номер базы. Учтите, что у базы может быть несколько номеров, и все они могут быть переданы в шифровке.

В последовательности первые 2 цифры означают длину полезной информации шифровки, то есть целое число:  $n$  ( $1 \leq n < 10^2$ ), третья цифра число  $k$  ( $1 \leq k \leq 5$ ) – длина номера секретной базы соответственно. Цифры в шифровке не разделяются пробелами. Определить целостность полученного сообщения, идентифицировать его и вывести номера секретных баз, если шифровка содержит номера секретных баз.

### Код программы на языке Си#.

### Визуализация работы программы.

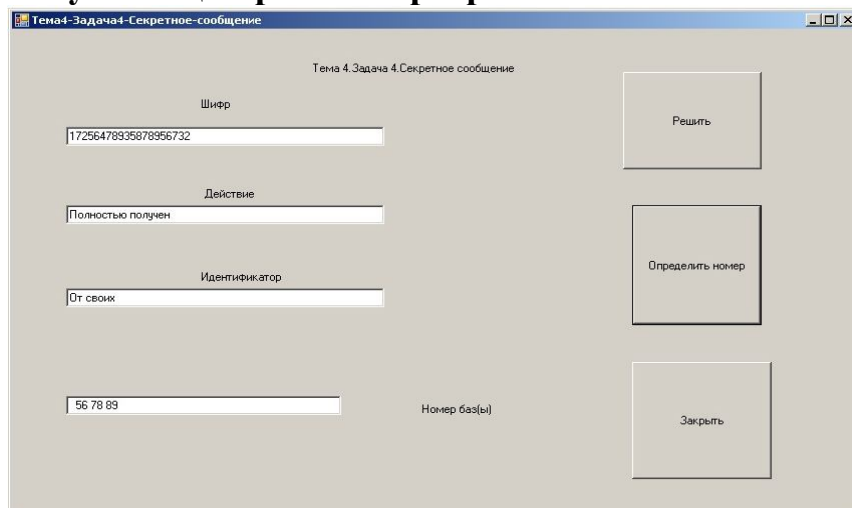


Рис. 16. Визуализация работы программы секретное сообщение.

## ТЕМА 5: СОРТИРОВКИ

### Цель.

Овладеть различными методами упорядочивания данных, отличающимися как по скорости, так и по используемой памяти. Умение оценивать сложность алгоритма.

### Основные понятия.

Сортировка – упорядочивание множества данных по определенному алгоритму.

### Ключевые слова.

Сортировка выбором, указатель, ссылка, сортировка слиянием.

### Задача1. Сортировка подсчетом и вставкой.

А) Сортировка вставкой. Массив вводится по элементам, и каждый

элемент сразу ставится на своё место в нужном порядке, т.е. ищется его место, начиная с конца, все большие элементы сдвигаются к концу.

Б) Сортировка подсчётом. В массиве большого размера известен небольшой диапазон изменения элементов. Упорядочить его по не убыванию, подсчитывая количество вхождений каждого значения в исходный массив. Используется вспомогательный массив для подсчёта.

#### **Код программы на языке Си++.**

```
#pragma once
.....
#pragma endregion
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e)
    {
        Form1::Close();
    }

private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e)
    {
        int a[1000];
        int b[1000];
        int c[1000];

        for(int i = 0;i<1000;i++)
        {
            a[i] = 0;
            b[i] = 0;
            c[i] = 0;
        }

        array<wchar_t>^mas=textBox1->Text->ToCharArray();

        int n = mas->Length;
        int l = 0;
int j = 0;
        String^h = "";

        for(int i = 0;i<n;i++)
        {
            if(Convert::ToString(mas[i])!=" ")
            {
                h = Convert::ToString(mas[i]);
                l = l*10 + Convert::ToInt16(h);
                if(i==n-1)
                {
                    a[j] = l;
                }
            }
            else
            {
                a[j] = l;
            }
        }
    }
}
```

```

        l=0;
        j++;
    }

    }
    h = "";

//Сортировка вставкой
int newEl, l;

for (int i = 0; i <= j; i++)
{
    newEl = a[i];
    l = i;
    while(l >= 0 && b[l] > newEl)
    {
        b[l+1] = b[l];
        l--;
    }
    b[l+1] = newEl;
}
for(int i = 0;i<=j;i++)
    h+=Convert.ToString(b[i+1])+" ";

textBox2->Text = h;

//Сортировка подсчетом
h="";
int max = 0;

for(int i = 0;i<=j;i++)
{
    c[a[i]]++;
    if(max<a[i])max = a[i];
}

for(int i = 0;i<=max;i++)
{
    for(int j = 1;j<=c[i];j++)
    {
        h+=Convert.ToString(i)+" ";
    }
}

textBox3->Text = h;
}
};
}

```

## Визуализация работы программы.

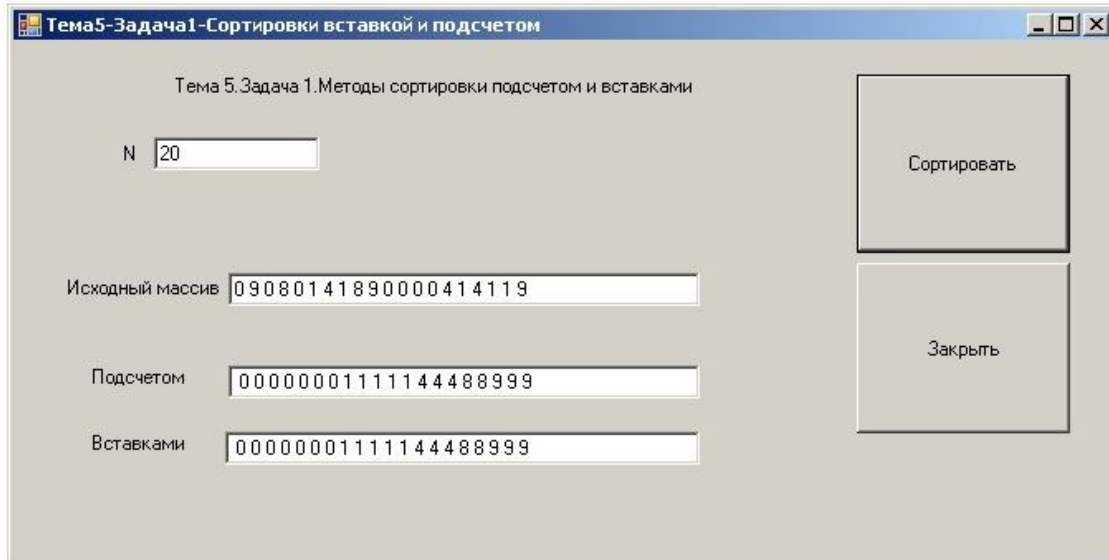


Рис. 17. Визуализация работы программы сортировка подсчетом и вставками.

### Задача2. Сортировка выбором и Quicksort.

А) Сортировка выбором.

Б) Быстрая сортировка Quicksort. На каждом этапе массив разбивается на две части относительно некоторого центрального значения: слева стоят все меньшие его, справа – все большие или равные. На каждую половинку рекурсивно запускаем аналогичную процедуру. Разбиение заканчивается, когда размер очередного фрагмента становится не больше 2. Тогда два элемента можно переставить в результате одного сравнения.

#### Код программы на языке Си++.

```
#pragma once
.....
#pragma endregion
private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e)
{
}
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e)
{
    Form1::Close();
}
public: void quickSort(int a[],int L,int R)
{
    int i = L, j = R;
    int temp;
    int sred = a[(L + R) / 2];

    while (i <= j) {
```

```

while (a[i] < sred)
i++;
while (a[j] > sred)
j--;
if (i <= j) {
temp = a[i];
a[i] = a[j];
a[j] = temp;
i++;
j--;
}
};

//Рекурсия
if (L < j)
quickSort(a, L, j);
if (i < R)
quickSort(a, i, R);
}
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e)
{
const int N=1000;
int a[N];
int b[N];
array<wchar_t>^mas=textBox1->Text->ToCharArray();
int l = 0;
String^h="";
int n = mas->Length;
int j = 0;

for(int i = 0;i<n;i++)
{
if(Convert::ToString(mas[i])!=" ")
{
h =Convert::ToString(mas[i]);
l = l*10 + Convert::ToInt16(h);
if(i==n-1)
{
a[j] = l;
}
}
else
{
a[j] = l;
l=0;
j++;
}
}
//Копирование массива a[] в массив b[]

```



```

for(int i = 0;i<=j;i++)
    b[i] = a[i];

//Сортировка выбором
for(int i = 0;i<=j;i++)
{
    int MinVal = i;
    for(int k = i + 1;k<=j;k++)
    {
        if(a[k]<a[MinVal])
            MinVal = k;
    }
    int temp = a[i];
    a[i] = a[MinVal];
    a[MinVal] = temp;
}
h = "";
for(int i = 0;i<=j;i++)
    h+=a[i]+" ";
textBox2->Text = h;
//Сортировка QuickSort
quickSort(b,0,j);
h = "";
for(int i = 0;i<=j;i++)
    h+=b[i]+" ";
textBox3->Text = h;
}
};
}

```

### Визуализация работы программы.

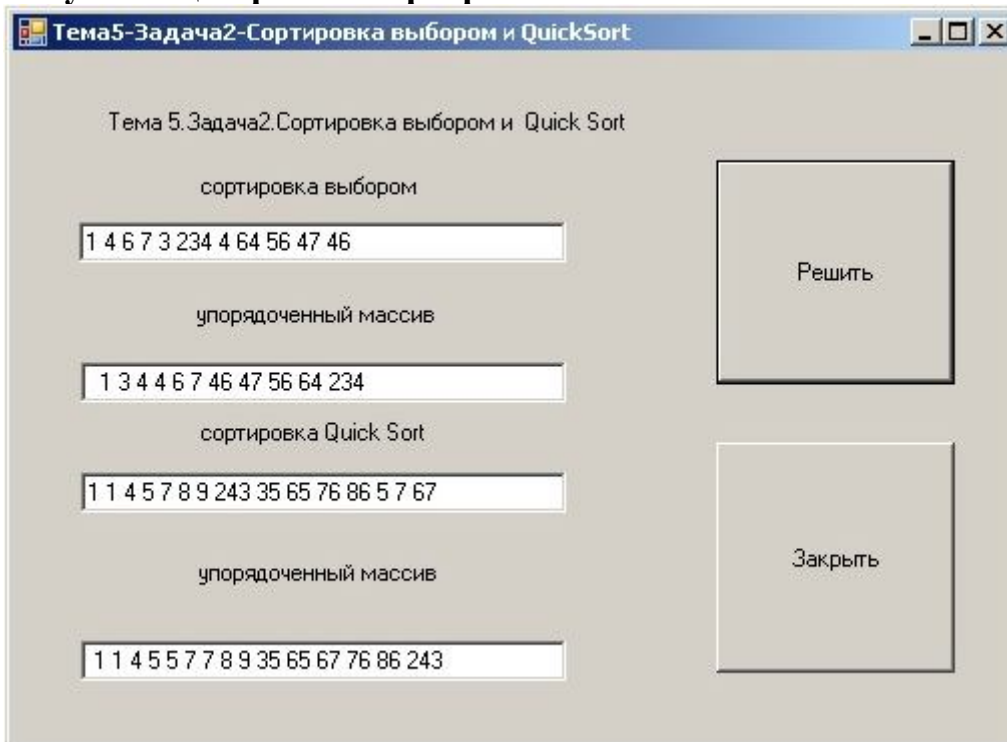


Рис. 18. Визуализация работы программы сортировка выбором и quicksort.

### Задача3. Указатели и ссылки

Написать программу, в которой используются следующие операции с указателями и ссылками:

- 1) Указатель на число
- 2) Указатель на массив (также и массив указателей)
- 3) Арифметика с указателями
- 4) Указатель на указатель

#### Код программы на языке Си#.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace za
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        unsafe int* swap(int* x, int* y)
        {
            int sum;
            sum = *x + *y;
            return &sum;
        }
        private void button1_Click(object sender, EventArgs e)
        {
            unsafe
            { //1
                int* p;
                int x = 8;
                p = &x;
                string h = " ";
                h = h + Convert.ToString((int)p) + " " + Convert.ToString(*p);
                textBox1.Text = h;
                //2
                int y = 4;
                int* m;
                m = &y;
                int* sum = swap(&x, &y);
                string h1 = " ";
                h1 = h1 + Convert.ToString(*sum);
                textBox2.Text = h1;
                //3
            }
        }
    }
}
```

```

string h2 = " ";
int[] a = new int[10];
for (int i = 0; i < 10; i++)
{
    a[i] = i;
}
fixed (int* k=&a[0])
{
    for (int i = 0; i < 10; i++)
    { h2 = h2 + " " + Convert.ToString(*(k + i)); }
    textBox3.Text = h2;
}
textBox3.Text = h2;
//4
int* s;
int w = 10;
s = &w;
int* r;
r = &w;
byte* z;
z = (byte*)r + 4;
string h3 = "";
h3 = Convert.ToString(*z);
textBox4.Text = h3;
//5
string h4="";
int** c = &m;
**c = **c + 30;
h4 = h4 + Convert.ToString(y) + " " + Convert.ToString(**c) + " " +
Convert.ToString((int)*c);
textBox5.Text = h4;
}
}
private void button2_Click(object sender, EventArgs e)
{
    Form1.ActiveForm.Close();
}
}
}

```

### **Код программы на языке Си++.**

```
#pragma once
```

```
.....
```

```
#pragma endregion
```

```
private: System::Void textBox3_TextChanged(System::Object^ sender,
System::EventArgs^ e) {
```

```
}
```

```
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^
e)
```

```
{
```

```
Form1::Close();
```

```
}
```

```
int*sum(int*a,int*b)
```

```

    {
        int sum = *a+*b;
        return &sum;
    }
    int*b;
    int n;

```

```

private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e)

```

```

    {
        //1 Указатель на переменную
        String^h = "";
        int x = 8;
        int*p = &x;
        h+=Convert::ToString((int)p)+ " " +Convert::ToString(*p);
        textBox1->Text = h;
        h = "";
        //2 Указатель аргумент функции
        int y = 5;
        int*m=&y;
        int*sum1 = sum(&x,&y);
        h=Convert::ToString(*sum1);
        textBox2->Text = h;
        h = "";
        //3 Указатель на массив
        int t = b[0];
        int*k = &t;
        for(int i = 0;i<n;i++)
            h+=Convert::ToString(*(k+i))+ " ";
        textBox3->Text = h;
        h = "";
        //4 Операции над указателями
        int w = 10;
        int*s = &w;
        int*r = &w;
        char* z;
        z=(char*)r + 4;
        h = Convert::ToString(*z);
        textBox4->Text = h;
        h = "";

        //5 Указатель на указатель
        int**c = &m;
        **c+=30;
        h+=Convert::ToString(**c)+ " " + Convert::ToString((int)*c);
        textBox5->Text = h;
    }

```

```

private: System::Void button3_Click(System::Object^ sender, System::EventArgs^ e)

```

```

    {
        n = Convert::ToInt16(textBox6->Text);
        b = new int[n];
    }

```

```

        textBox7->Text = "Массив создан!";
    }
private: System::Void button4_Click(System::Object^ sender, System::EventArgs^ e)
{
    delete []b;
    textBox7->Text = "Массив удалён!";
}
private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e)
{
}
};
}

```

### Визуализация работы программы.

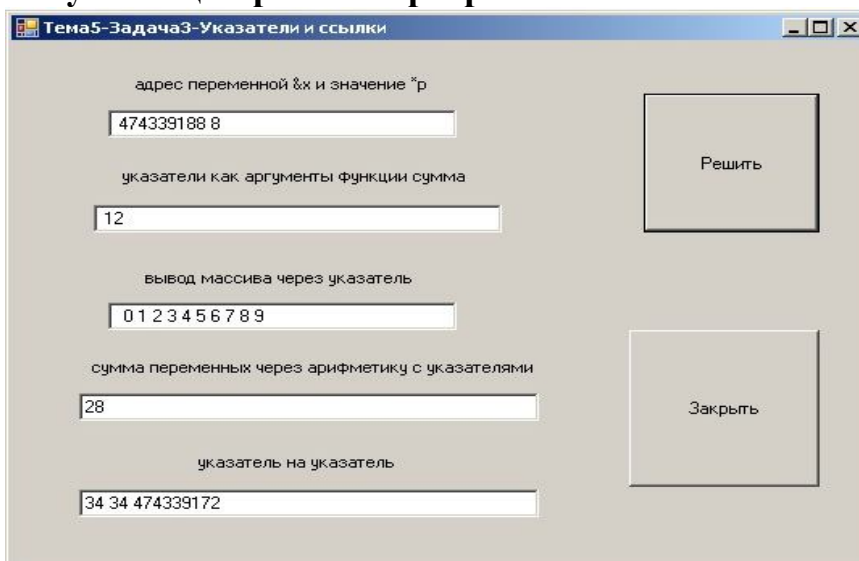


Рис. 19. Визуализация работы программы работы с указателями.

### Задача4. Сортировка слиянием.

На каждом этапе большой массив разбивается на две части, каждая из которых отдельно упорядочивается (сортируется). Потом обе части снова сливаются в единый (уже упорядоченный) массив за один проход. К каждой части применяется та же процедура. Разбиение продолжается до тех пор, пока в массиве не останется только два элемента, которые упорядочиваются за одно сравнение.

#### Код программы на языке Си#.

```

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 Сортировка_слиянием

```

```

{
public partial class Form1 : Form
{
public Form1()
{
InitializeComponent();
}
void Merge(int []mas,int first,int last)
{
int middle, start, final, j;
int[] A = new int[100];

middle = (first + last) / 2;
start = first;
final = middle + 1;
for(j= first;j<= last;j++)

if((start<=middle)&&((final>last)||((mas[start]<mas[final])))
{
A[j] = mas[start];
start++;
}
else
{
A[j] = mas[final];
final++;
}
for (j = first; j <= last; j++)
mas[j] = A[j];
}
void MergeSort(int []mas,int first,int last)
{
if(first<last)
{
MergeSort(mas, first, (first + last) / 2);
MergeSort(mas, (first + last) / 2+1, last);
Merge(mas, first, last);
}
}
private void button6_Click(object sender, EventArgs e)
{
int N = Convert.ToInt32(textBox6.Text);
int i;
int[] mas = new int[N];
string h2 = " ";
Random rnd = new Random();
for (i = 0; i < N; i++)
{
mas[i] = rnd.Next(100);
h2 = h2 + " " + Convert.ToString(mas[i]);
}
textBox4.Text = h2;
}
}

```

```

MergeSort(mas, 0, N-1);
string h = " ";
for (int k = 0; k < N; k++)
    h = h + " " + Convert.ToString(mas[k]);
textBox5.Text = h;
}
private void button5_Click_1(object sender, EventArgs e)
{
    Form1.ActiveForm.Close();
}
}
}

```

### Визуализация работы программы.

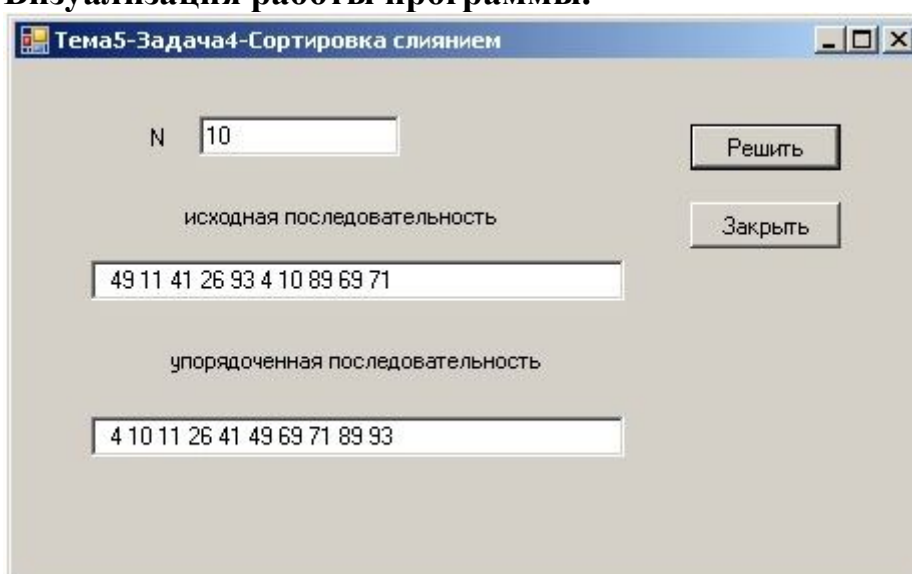


Рис. 20. Визуализация работы программы сортировки слиянием.

## ТЕМА 6: СИМВОЛЬНЫЕ СТРОКИ (ТЕКСТЫ)

### Цель.

Изучение особенностей массивов, содержащих символьную информацию (строки, тексты).

### Основные понятия.

Во многих языках программирования для работы с текстами (символьными строками) используются массивы типа `char` со специальным символом конца `'\0'` или специальные классы типа `string`. Для них разработано много специальных функций и методов классов. Приведём некоторые из них для символьных массивов:

`int strlen (char *)` – функция вычисляет длину строки

`char * strcpy (char *, char *)` – копирует вторую строку по первому адресу (массивы нельзя присваивать)

`int strcmp (char *, char *)` – сравнивает две строки

`char * strcat (char *, char *)` – присоединяет вторую строку в конец первой  
Обычно тексты обрабатываются по символам или по словам (группам символов).

Для ввода символьных строк используют специальные функции, т.к. пробелы играют особую роль и рассматриваются в стандартных функциях и операциях как разделители. При выводе символьных строк можно воспользоваться и стандартными операциями и функциями.

`gets (char *)`  
`cin.getline (char *, int)`

### **Ключевые слова.**

Строка, текст, слово, символ, нулевой символ, адрес, указатель

### **Задача1. Подсчет слов в тексте.**

Написать программу, которая вычислит и напечатает количество слов в заданном тексте (взяв текст из окна и выведя его в текстовый файл). Текст состоит из последовательности слов, разделённых пробелами. Слово содержит произвольные символы (буквы), отличные от пробела. Слово не является частью другого слова и не содержит в себе другие слова. За исходный текст взять алгоритм, представленный ниже.

### **Алгоритм.**

Будем последовательно выделять каждое слово и вести их подсчёт. Начало слова определяется как не пробел, конец слова находится по пробелу или концу текста. Схема обработки слов может выглядеть так:

ЦИКЛ до конца текст – будет найдено и обработано ровно одно слово  
НАЙТИ начало слова (в цикле пропустить все пробелы)  
ЕСЛИ достигли конца текста, ТО ВЫХОД из цикла  
ЗАПОМНИТЬ позицию начала слова  
НАЙТИ конец слова (в цикле пропустить все не пробелы)  
ЗАПОМНИТЬ позицию конца слова  
ОБРАБОТАТЬ слов – зависит от конкретной задачи  
КОНЕЦ\_ЦИКЛА  
ПЕЧАТЬ результатов

### **Решение.**

Согласно приведённому алгоритму напишем программу. Наша программа будет содержать большой цикл по словам и внутри него два цикла для поиска начала и конца слова. В зависимости от задачи циклы могут быть и в пункте ОБРАБОТАТЬ СЛОВО.

### **Код программы на языке Си++.**

```
#pragma once  
.....  
#pragma endregion
```



```

private: System::Void button3_Click(System::Object^ sender, System::EventArgs^ e)
    {
        Form1::Close();
    }
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e)
    {
        StreamWriter^sw;
        StreamReader^sr;
        String^h = richTextBox1->Text;
        FileInfo^fi = gcnew FileInfo(Application::StartupPath+"\\slova.txt");
        if(fi->Exists)
        {
            sw = fi->AppendText();
            sw->Close();
        }
        else
        {
            sw=fi->CreateText();
            sw->WriteLine(h);
            sw->Close();
        }
        sr = gcnew StreamReader(Application::StartupPath+"\\slova.txt");
        h = sr->ReadToEnd();
        sr->Close();
        richTextBox1->Text = Convert::ToString(h);
    }
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e)
    {
        array<wchar_t>^mas = richTextBox1->Text->ToCharArray();
        int n = mas->Length;
        int k = 1;
        for(int i = 1;i<n;i++)
            if(mas[i-1] == ' ' && mas[i]!=' ')
                k++;
        textBox1->Text = Convert::ToString(k);
    }
private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e)
    {
    }
};
}

```

## Визуализация работы программы.

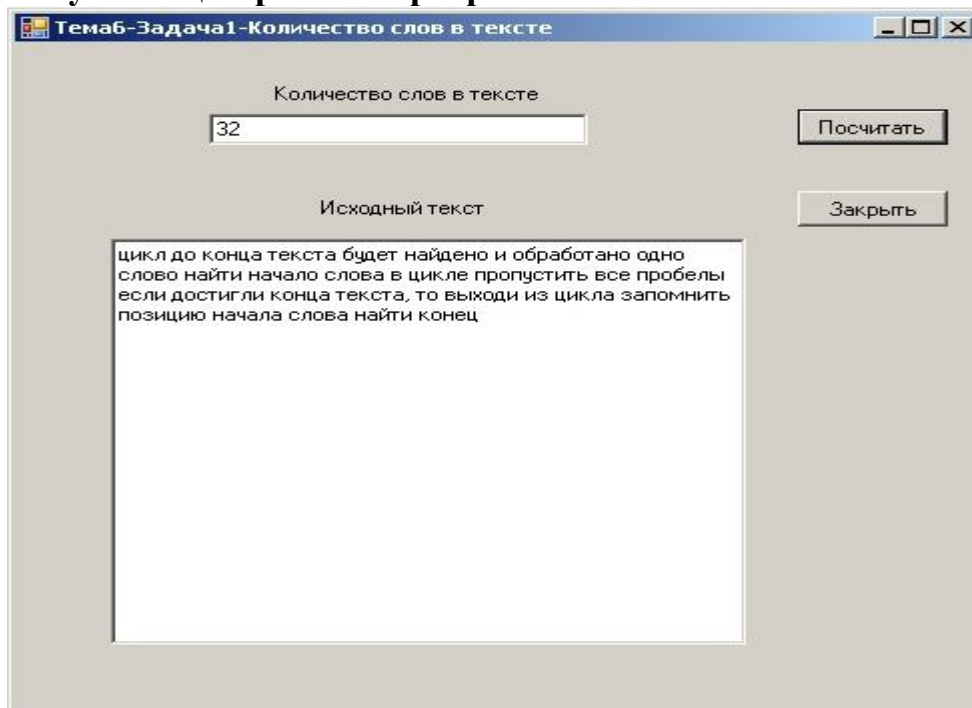


Рис. 21. Визуализация работы программы количество слов.

### Задача2. Большие и малые буквы в тексте.

Взять текст из текстового файла. Заменить в этом тексте все строчные буквы на прописные и наоборот. Показать исходный текст и полученный текст в окнах на форме. Далее вывести полученный текст в текстовый файл.

Исходные данные: Пример Простого Текста

Результат: ПРИМЕР ПРОСТОГО ТЕКСТА

### Код программы на языке Си#.

```
using System;
using System.IO;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace VanAnd_T6_Z2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        String[] a = new String[100];
        private void button1_Click(object sender, EventArgs e)
```

```

    {
        FileStream file1 = new FileStream("slova.txt", FileMode.Open,
        FileAccess.Read);
        StreamReader rd = new StreamReader(file1);
        String s;

        int p = 0;
        do
        {
            s = rd.ReadLine();
            if (s != null)
            {
                a[p++] = s;
            }
        } while (s != null);
        richTextBox1.Lines = a;
        file1.Close();
        rd.Close();
    }
    private void button2_Click(object sender, EventArgs e)
    {
        String s = "";
        String h = "";

        for (int i = 0; a[i]!=null; i++)
        {
            s = a[i];
            char[] mas = s.ToCharArray();
            for (int k = 0; k < mas.Length;k++ )
            {
                if (mas[k] <= 'Z')
                    mas[k] = (char)(mas[k] + 32);
                else if(mas[k]<'A')
                    mas[k] = (char)(mas[k] - 32);
                else if(mas[k]<='Я')
                    mas[k] = (char)(mas[k] + 32);
                else
                    mas[k] = (char)(mas[k] - 32);

                h += mas[k];
            }
        }
        richTextBox2.Text = h;
        FileStream file1 = new FileStream("slova.txt", FileMode.Create,
        FileAccess.Write);
        StreamWriter wr = new StreamWriter(file1);
        wr.WriteLine(h);
        wr.Close();
        file1.Close();
    }
    private void button3_Click(object sender, EventArgs e)
    {

```

```

        Form1.ActiveForm.Close();
    }
    private void button4_Click(object sender, EventArgs e)
    {
        String h = richTextBox1.Text;
        FileStream file1 = new FileStream("slova.txt", FileMode.Create,
        FileAccess.Write);
        StreamWriter wr = new StreamWriter(file1);
        wr.WriteLine(h);
        wr.Close();
        file1.Close();
    }
    private void Form1_Load(object sender, EventArgs e)
    {
    }
}
}
}

```

### Визуализация работы программы.

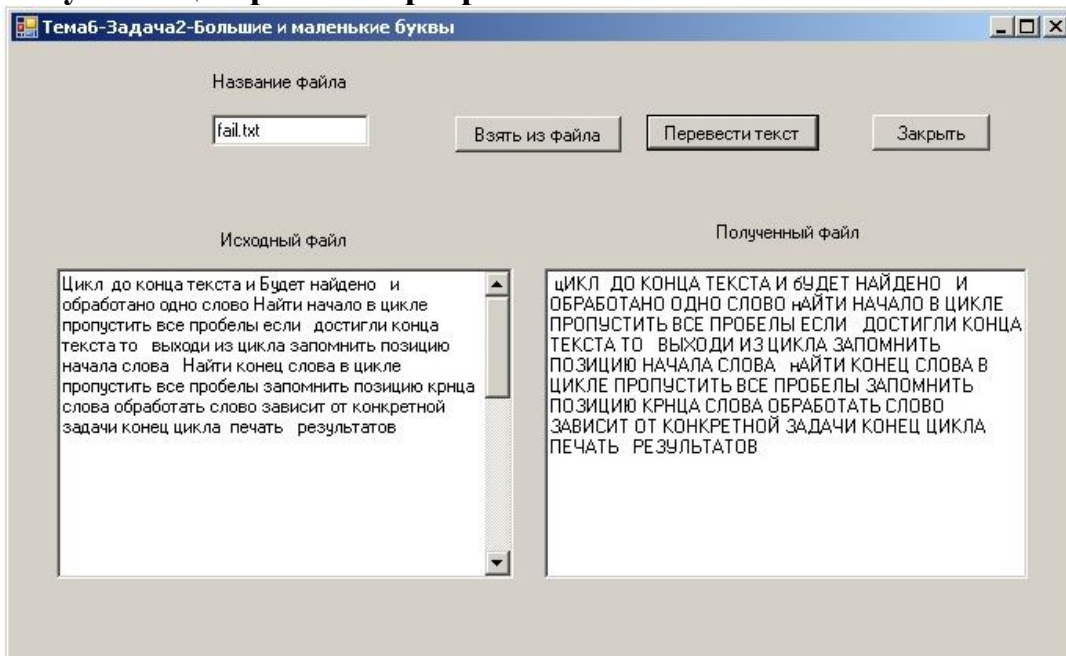


Рис. 22. Визуализация работы программы большие и малые буквы.

### Задача3. Удаление лишних пробелов в тексте.

Дан текст, состоящий из слов. Напечатать его, оставляя между словами ровно по одному пробелу.

Исходные данные: Пример простого текста

Результат: Пример простого текста

### Код программы на языке Си++.

## Визуализация работы программы.

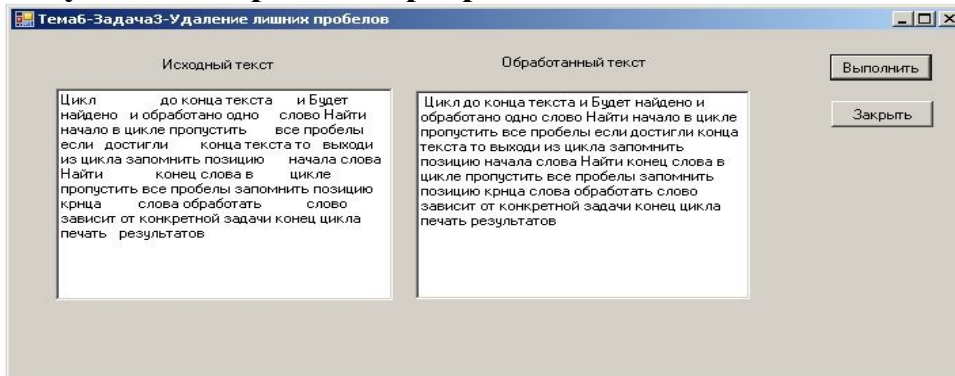


Рис. 23. Визуализация работы программы удаление пробелов.

### Задача4. Проверка правильности математической формулы.

Дана символьная строка, содержащая некоторую формулу с круглыми скобками. Написать программу, которая напечатает «ОК», если скобки расставлены правильно, и «НЕТ» в противном случае.

Исходные данные:  $(a+b)*(c-)$

Результат: ОК

Код программы на языке Си++.

## Визуализация работы программы.

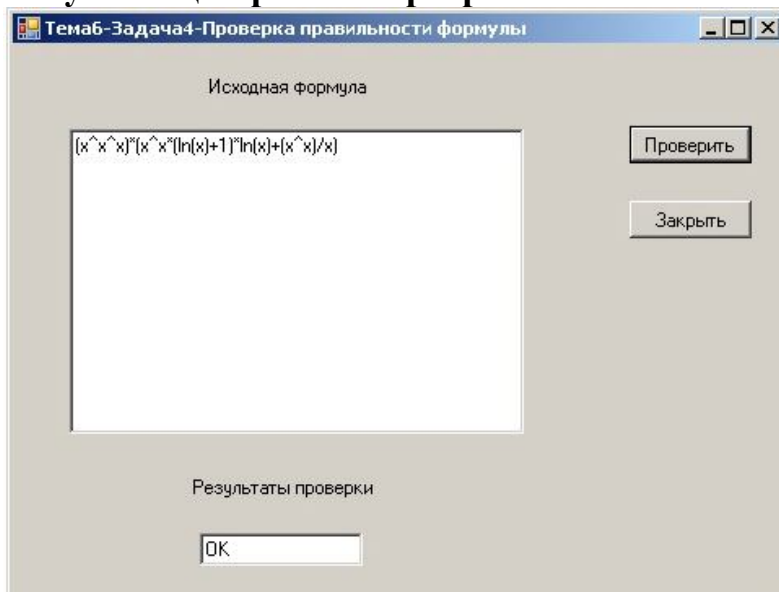


Рис. 24. Визуализация работы программы правильность формулы.

## ТЕМА 7: ДВУМЕРНЫЕ МАССИВЫ (МАТРИЦЫ)

### Цель.

Изучение алгоритмов обработки многомерных массивов или их фрагментов.

### Основные понятия.

Описание и выделение памяти для автоматических и динамических матриц  
Работа с элементами

### **Ключевые слова.**

Матрица, многомерный массив, строка, столбец, определитель

### **Задача1. Транспонирование матрицы.**

Дана квадратная матрица целых чисел порядка  $n$ . Транспонировать её относительно главной диагонали и напечатать. Другие массивы не использовать. Значения элементов не превосходят  $10^5$ . Вывести исходную и транспонированную матрицы на консоль и в текстовый файл.

### **Алгоритм.**

Обычные действия по описанию операции транспонирования (бегать по всем строкам и столбцам матрицы) приводят к тому, что в результате получается та же матрица, что и исходная, т.к. она транспонируется дважды. Главная диагональ разбивает матрицу на 2 треугольные части и будем бегать по одной из них: в приведённом примере – по нижней.

### **Код программы на языке Си++ (консольное приложение Win32).**

```
#include "stdafx.h"
#include <stdio.h>
#include <fstream>
#include <iostream>

using namespace System;
using namespace std;

int main(array<System::String ^> ^args)
{
    setlocale(LC_ALL,"Russian");
    int N = 0;
    printf("Введите N = ");
    scanf("%d",&N);
    printf("\n");

    int**b = new int*[N+1];
    for(int j = 0;j<N+1;j++)
        b[j] = new int[N+1];

    for(int i = 0;i<N;i++)
    {
        for(int j = 0;j<N;j++)
        {
            cout<<"Введите элемент матрицы a["<<i+1<<","<<j+1<<"]="";
            cin>>b[i][j];
        }

        cout<<endl;
    }
    //Вывод исходной матрицы в консоль
```

```

cout<<"Исходная матрица:";
for(int i = 0;i<N;i++)
{
    cout<<endl;
    for(int j = 0;j<N;j++)
    {
        cout<<" "<<b[i][j]<<" ";

    }
}
//Вывод исходной матрицы в файл
ofstream fout("transpmatr.txt");
fout<<" Исходная матрица: \n";
for(int i = 0;i<N;i++)
{
    for(int j = 0;j<N;j++)
        fout<<" "<<b[i][j]<<" ";
    fout<<endl;
}
//Транспонирование матрицы
for(int i = 0;i<N;i++)
{
    for(int j = i;j<N;j++)
    {
        int temp = b[i][j];
        b[i][j] = b[j][i];
        b[j][i] = temp;
    }
}
//Вывод траспонированной матрицы в консоль
cout<<"\n\nТранспонированная матрица:";
for(int i = 0;i<N;i++)
{
    cout<<endl;
    for(int j = 0;j<N;j++)
    {
        cout<<" "<<b[i][j]<<" ";

    }
}
//Вывод транспонированной матрицы в файл
fout<<" Транспонированная матрица: \n";
for(int i = 0;i<N;i++)
{
    for(int j = 0;j<N;j++)
        fout<<" "<<b[i][j]<<" ";
    fout<<endl;
}
fout.close();

//Взятие из файла
char buff[50];

```

```

ifstream fin("transpmatr.txt");
if(!fin.is_open())
    cout<<"Файл не может быть открыт";
else
{
    fin>>buff;
    fin.close();
    cout<<buff<<endl;
}
system("pause");
return 0;

```

### }Визуализация работы программы.

```

D:\Алемасов2016\ме тодички-2016-оп\Ванюков Андрей\tema7\VanAnd_T7_Z1\Debug\Van...
Введите N = 3
Введите элемент матрицы a[1,1]=4
Введите элемент матрицы a[1,2]=5
Введите элемент матрицы a[1,3]=6
Введите элемент матрицы a[2,1]=3
Введите элемент матрицы a[2,2]=4
Введите элемент матрицы a[2,3]=7
Введите элемент матрицы a[3,1]=5
Введите элемент матрицы a[3,2]=6
Введите элемент матрицы a[3,3]=8
Исходная матрица:
4 5 6
3 4 7
5 6 8
Транспонированная матрица:
4 3 5
5 4 6
6 7 8 Исходная
Для продолжения нажмите любую клавишу . . .

```

Рис. 25. Визуализация работы программы транспонирование матрицы.

### Задача2. Определитель матрицы.

Написать программу, которая будет вычислять определитель матрицы порядка N. В консоли задаётся размер матрицы N, и сама матрица.

#### Код программы на языке Си++ (консольное приложение CLR).

```

#include "stdafx.h"

//#include "stdlib.h"

using namespace System;

int ** get_mem_sq_matrix(int);
void mout (int **, int);
int det(int **, int, int **);
int ** crm (int **, int, int, int **);
int in () {
    String^ inp = Console::ReadLine();
    return int::Parse(inp);
}

```



```

void out (int a) {
    Console::WriteLine(a.ToString());
}
int ** l;
int main(array<System::String ^> ^args)
{
    Console::WriteLine(L"Размерность массива N: ");
    /*String^ inp = Console::ReadLine();
    int n = int::Parse(inp);*/
    int n = in();
    int ** m = get_mem_sq_matrix(n);
    l = get_mem_sq_matrix(n);

    Console::WriteLine(L"Массив, по элементу на одной строке: ");
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            m [i][j] = in();
        }
    }
    Console::WriteLine(L"Исходный массив: ");
    mout(m, n);
    double dt;
    dt = det (m, n, l);
    Console::WriteLine();
    Console::WriteLine(L"det = " + dt);

    delete[] m;
    delete[] l;
    Console::ReadLine();
    //system("pause");
    return 0;
}
int ** get_mem_sq_matrix(int n) {
    int ** a;
    a = new int * [n];
    for (int i = 0; i < n; i++) {
        a[i] = new int [n];
    }
    return a;
}
void mout(int ** m, int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            Console::Write(m[i][j] + " ");
        }
        Console::Write("\n");
    }
}
int det (int ** m, int n, int ** l) //нахождение определителя
{
    int d = 0;

```

```

    if (n == 1) {
        d = m[0][0];
    } else {
        for (int i = 0, z = 1; i < n; i++, z = z * (-1)) {
            d += z * m[0][i] * det(crm(m, n - 1, i, l), n-1, l);
        }
    }
    return d;
}
int ** crm (int** m, int n, int c, int ** l) {
    for (int i = 0; i < n; i++) {
        for (int j = 0, st = 0; j < n + 1; j++) {
            if (j != c) {
                l[i][st++] = m[i + 1][j];
            }
        }
    }
    return l;
}
}

```

### Визуализация работы программы.

```

c:\ D:\lemasov2016\методички-2016-оп\Ванюков Андрей\tema7\VanAnd_T7_Z2\Debug\Van...
Введите размер матрицы N=3
Введите элемент матрицы a[1,1]=2
Введите элемент матрицы a[1,2]=3
Введите элемент матрицы a[1,3]=4
Введите элемент матрицы a[2,1]=5
Введите элемент матрицы a[2,2]=6
Введите элемент матрицы a[2,3]=7
Введите элемент матрицы a[3,1]=8
Введите элемент матрицы a[3,2]=7
Введите элемент матрицы a[3,3]=6
Исходная матрица:
 2  3  4
 5  6  7
 8  7  6
Определитель матрицы Det = 0
Для продолжения нажмите любую клавишу . . .

```

Рис. 26. Визуализация работы программы определитель матрицы.

### Задача2. Обратная матрица.

Написать программу, которая будет вычислять обратную матрицу, если она существует. В окне задаётся матрица  $A$ , то есть построчно записана сама матрица целых чисел  $A$ , элементы в строке разделены одним пробелом. Вычислить и напечатать обратную матрицу  $B$ , т.е. квадратную матрицу порядка  $n$  такую, что произведение  $A*B = E$  (где  $E$  – единичная матрица). Если обратная матрица не существует, напечатать -1. Также показать матрицу произведения  $A*B$  (проверить, что матрица  $B$  обратная).

### Код программы на языке Си++.

```

#pragma once
.....
#pragma endregion
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^
e)
    {
        Form1::Close();
    }
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^
e)
    {
        richTextBox2->Text = "";
        richTextBox3->Text = "";
        array<wchar_t>^mas = richTextBox1->Text->ToCharArray();
        int n = mas->Length;
        int m = 0;
        int l = 0;
        int c[1000];
String^h="";
int j = 0;

        for(int i = 0;i<n;i++)
        {
            if(Convert::ToString(mas[i])!=" "&&mas[i]!='\n')
            {
                h =Convert::ToString(mas[i]);
                l = l*10 + Convert::ToInt16(h);
                if(i==n-1)
                {
                    c[j] = l;
                }
            }
            }else
            {
                c[j] = l;
                l=0;
                j++;
            }
        }

        for(int i = 0;mas[i]!='\n';i++)//нахождение размера будущей
матрицы
            if(mas[i]!=' ')
                m++;
        double**a = new double*[m];
        for(int i = 0;i<m;i++)
            a[i] = new double[2*m];
        int k = 0;
        for(int i = 0;i<m;i++)
        {
            for(int j=0;j<m;j++)

```

```

        {
            a[i][j] = c[k];
            k++;
        }
    }
    double**b = new double*[m];
    for(int i = 0;i<m;i++)
        b[i] = new double[m];

    for(int i = 0;i<m;i++)
        for(int j = 0;j<m;j++)
            b[i][j]=a[i][j];
    for(int i = 0;i<m;i++)
        for(int j = m;j<2*m;j++)
            if(i==j-m)
                a[i][j] = 1;
            else a[i][j] = 0;
    for(int i = 0;i<m;i++)
    {
        double d = a[i][i];
        for(int z = 0;z<2*m;z++)
        {
            a[i][z]=a[i][z]/d;
        }

        for(int k = i+1;k<m;k++)
        {
            double el = a[k][i];
            for(int j = i;j<2*m;j++)
            {
                a[k][j]-=a[i][j]*el;
            }
        }
    }

    for(int i = m-1;i>0;i--)
    {
        for(int k = i-1;k>=0;k--)
        {
            double el = a[k][i];

            for(int j = i;j<2*m;j++)
            {
                a[k][j]-=a[i][j]*el;
            }
        }
    }

```

```

        }
    }
    for(int i = 0;i<m;i++)
    {
        for(int j = m;j<2*m;j++)
            richTextBox2->Text+=Convert::ToString(Math::Round(a[i][j],2))+" ";
        richTextBox2->Text+="\n";
    }
    for(int i = 0;i<m;i++)
        for(int j = 0;j<m;j++)
            a[i][j] = 0;
    for(int i = 0;i<m;i++)
        for(int j = 0;j<m;j++)
        {
            for(int k = 0;k<m;k++)
                a[i][j] += b[i][k]*a[k][j+m];
        }
    for(int i = 0;i<m;i++)
    {
        for(int j = 0;j<m;j++)
            richTextBox3->Text+=Convert::ToString(Math::Round(a[i][j],2))+" ";
        richTextBox3->Text+="\n";
    }
    for(int i = 0;i<m;i++)
    {
        delete[] b[i];
        delete[] a[i];
    }
    delete[] b;
    delete[] a;
}
private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e)
{
}
};
}

```

#### **Задача4. Решение системы уравнений методом Гаусса.**

Написать программу, которая будет вычислять решение системы линейных уравнений методом Гаусса, если оно существует. В окне задаётся матрица исходной системы уравнений и матрица свободных членов Все числа разделены одним пробелом. Вычислить и напечатать решение этой системы. Если решение не существует, напечатать -1. Также проверить решение системы уравнений.

## Визуализация работы программы обратной матрицы.

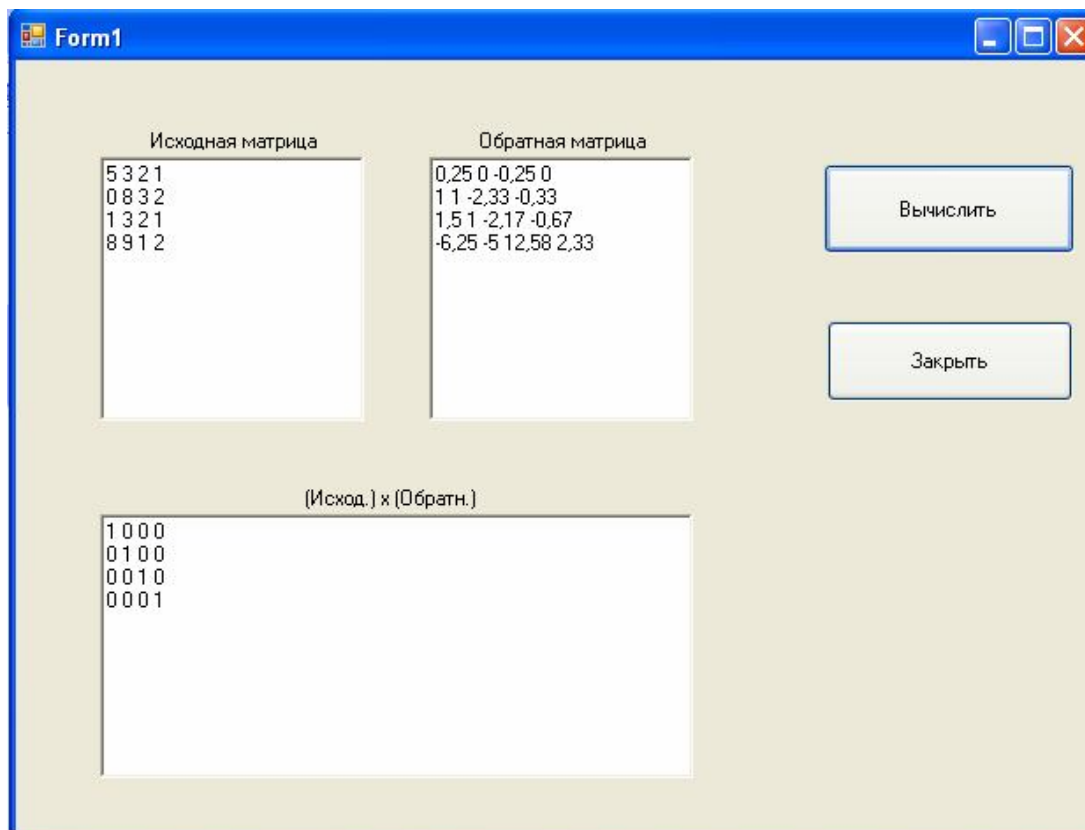
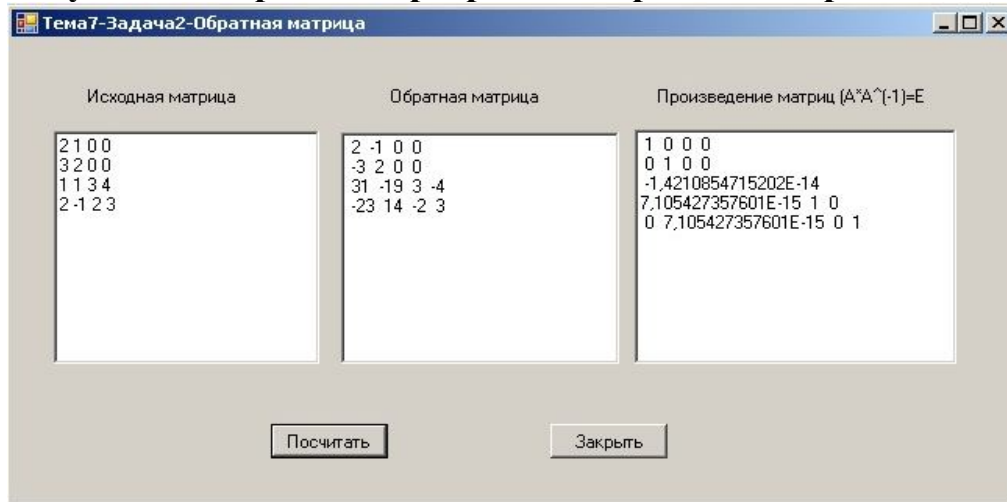


Рис. 27. Визуализация работы программы обратная матрица.

### Код программы на языке Си# (решение системы уравнений).

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
```

```
namespace VanAnd_T7_Z4
```

```

{
public partial class Form1 : Form
{
public Form1()
{
InitializeComponent();
}
private void Form1_Load(object sender, EventArgs e)
{
}
private void button1_Click(object sender, EventArgs e)
{

char[] ch = richTextBox1.Text.ToCharArray();
int n = ch.Length;
int m = 0;
int l = 0;
int[] c = new int[1000];
String h = "";
int j = 0;
for(int i = 0;i<n;i++)
{
if(Convert.ToString(ch[i])!=" "&&ch[i]!='\n')
{
h =Convert.ToString(ch[i]);
l = l*10 + Convert.ToInt16(h);
if(i==n-1)
{
c[j] = l;
}
}
else
{
c[j] = l;
l=0;
j++;
}
}
for(int i = 0;ch[i]!='\n'&&ch[i]!='\0';i++)//нахождение размера
будущей матрицы
if(ch[i]!=' ')
m++;
double[,] a = new double[m-1, m ];
double[,] aa = new double[m - 1, m];
//Заполнение массива a[] из массива c[]
int k = 0;
for (int i = 0; i < m-1; i++)
for (int l = 0; l < m; l++)
{
a[i,l] = c[k];
aa[i, l] = c[k];
}
}
}
}

```

```

        k++;
    }

    for (int i = 0; i < m-1; i++)
    {
        double d = a[i,i];
        for (int z = 0; z < m; z++)
        {
            a[i,z] = a[i,z] / d;
        }
        //Элементарные преобразования
        for (int r = i + 1; r < m-1; r++)
        {
            double el = a[r,i];
            for (int l = i; l < m; l++)
            {
                a[r,l] -= a[i,l] * el;
            }
        }
    }
    //Массив решений
    //Вычисление корней
    double[] x = new double[m-1];
    for (int i = 0; i < m - 1; i++)
        x[i] = a[i, m - 1];
    for (int i = m-2; i >= 0; i--)
    {
        for (int q = i+1; q < m-1; q++)
        {
            x[i] -= a[i, q]*x[q];
        }
    }

    textBox1.Clear();
    for (int i = 0; i < m - 1; i++)
        textBox1.Text += Convert.ToString(Math.Round(x[i],2)) + " ";
    //Массив свободных членов
    //Проверка корней
    double[] b = new double[m - 1];
    for (int i = 0; i < m - 1; i++)
        for (j = 0; j < m - 1; j++)
        {
            b[i] += aa[i, j] * x[j];
        }
    textBox2.Clear();
    for (int i = 0; i < m - 1; i++)
        textBox2.Text += Convert.ToString(b[i]) + " ";
}
private void button2_Click(object sender, EventArgs e)
{
    Form1.ActiveForm.Close();
}
}

```



}  
}  
**Визуализация работы программы.**

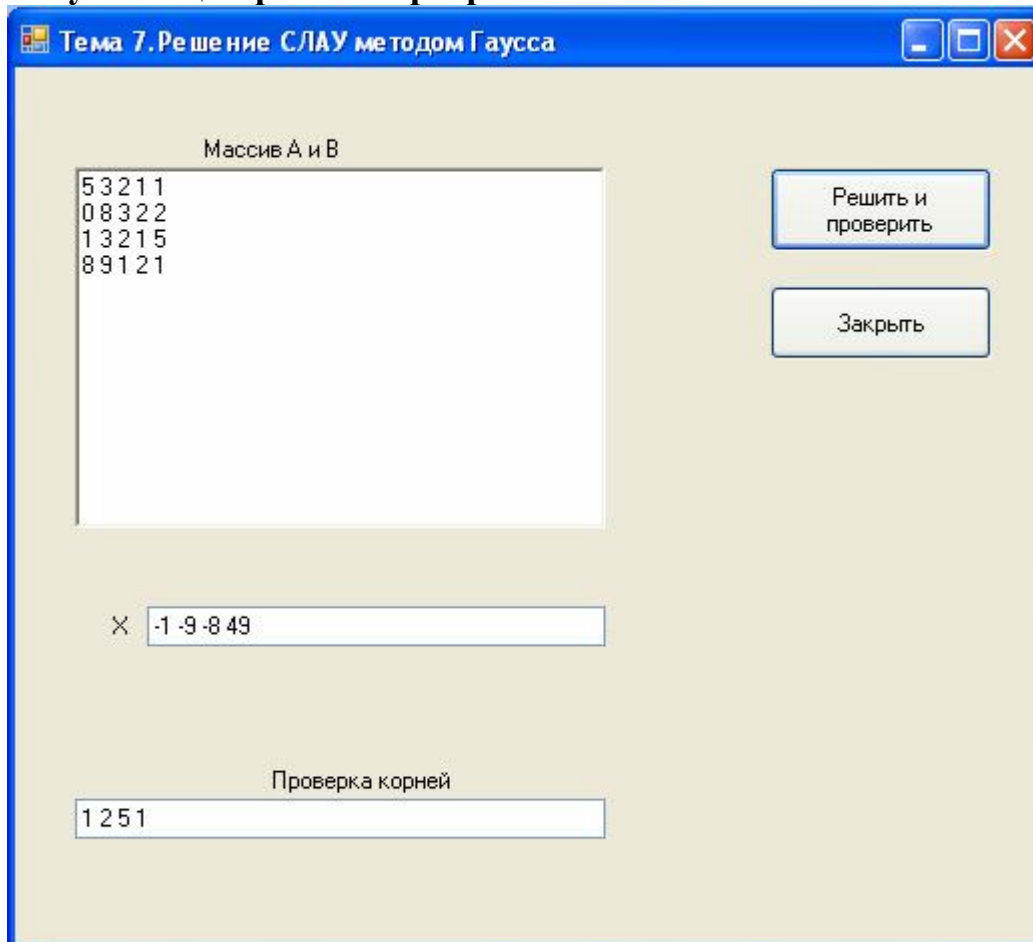


Рис. 28. Визуализация работы программы решение системы уравнений.

## ТЕМА 8: ФУНКЦИИ И РЕКУРСИИ

### Цель.

Получить навыки оформления одинаковых или почти одинаковых (похожих) фрагментов программы в виде процедур и функций. Научиться передавать нужные параметры разных типов и получать результаты, даже если их много.

### Основные понятия.

Описание и определение функции

Передача параметров-значений, адресов и ссылок

Передача массивов, матриц и указателей

Передача параметров в главную функцию

Описание функций с прямой и косвенной рекурсией

### Ключевые слова.

Функция, параметр, изменяющийся параметр, вызов функции, возврат одного результата и нескольких результатов, рекурсия

### **Замечания.**

Для демонстрации правильности работы той или иной функции (в этом и последующих разделах) необходимо в программу включать основную функцию и несколько вспомогательных функций.

### **Задача1. Функция n!.**

Написать функцию (с рекурсией и без) с одним параметром (целое n) для вычисления факториала n! В главной функции продемонстрировать вызовы функций и получение результатов.

#### **Алгоритм.**

Будем писать функцию согласно определению факториала.

#### **Код программы на языке Си++.**

```
#pragma once
.....
#pragma endregion
int F(int n )
{
    if(n<1)
        return 1;
    else
        return n*F(n-1);
}

int Fact(int n)
{
    int f = 1;
    for(int i =1;i<=n;i++)
    {
        f*=i;
    }
    return f;
}

private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e)
{
    int n = Convert::ToInt32(textBox1->Text);
    textBox2->Text = Convert::ToString(F(n));
    textBox3->Text = Convert::ToString(Fact(n));
}

private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e)
{
    Form1::Close();
}

};
}
```

## Визуализация работы программы.

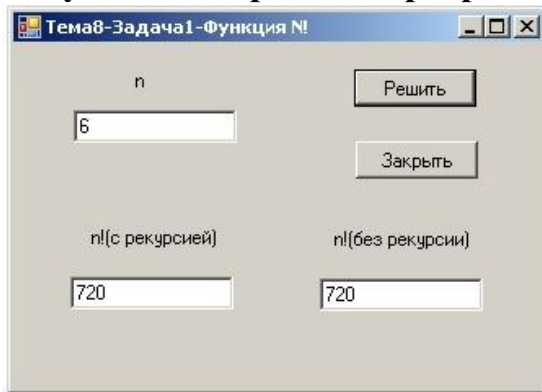


Рис. 29. Визуализация работы программы вычисления функции  $n!$ .

### Задача2. Функция вычисления числа сочетаний из $n$ по $k$ ( $C_n^k$ ).

Написать рекурсивную функцию `long long int CombiR (int, int)` и не рекурсивную `long long int Combi (int, int)` с двумя параметрами для вычисления числа сочетаний из  $n$  по  $k$ . Результат типа `long long int (__int64)`.

#### Код программы на языке Си++.

```
#pragma once
.....
#pragma endregion
int CknRec(int k,int n)
{
    if(k<1)
        return 1;
    else
        return n*CknRec(k-1,n-1)/k;
}

unsigned long long Ckn(int k,int n)
{
    unsigned long long f = 1;
    unsigned long long z = 1;
    unsigned long long x = 1;
    for(int i = 1;i<=k;i++)
        f*=i;
    for(int i = 1;i<=n;i++)
        z*=i;
    for(int i = 1;i<=n-k;i++)
        x*=i;
    return z/f/x;
}

private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e)
{
    Form1::Close();
}

private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e)
```

```

    {
        int k = Convert.ToInt32(textBox1->Text);
        int n = Convert.ToInt32(textBox2->Text);
        textBox3->Text = Convert.ToString(CknRec(k,n));
        textBox4->Text = Convert.ToString(Ckn(k,n));
    }
private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e)
{
    }
};
}

```

### Визуализация работы программы.

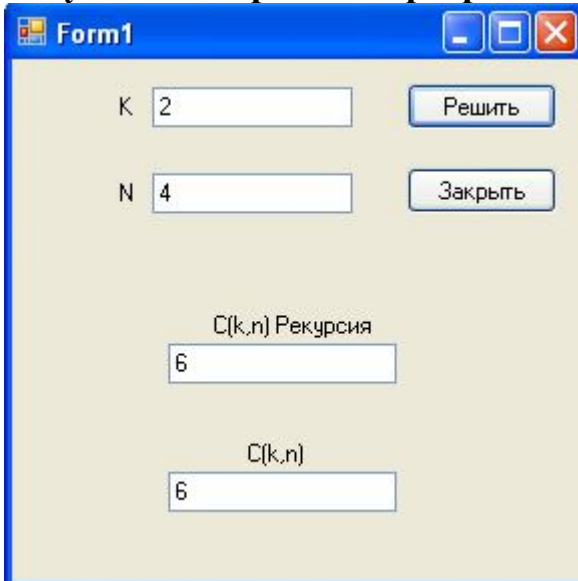


Рис. 30. Визуализация работы программы вычисления функции  $C_n^k$ .

### Задача3. Функция вычисления интеграла методом Симпсона.

Написать функцию вычисления интеграла по формуле Симпсона и методом прямоугольников. Имеется описание непрерывной на отрезке  $[a, b]$  функции (например  $\sin(x)$ ). Сравнить два метода решения. В окне задаются параметры – два вещественных числа – концы отрезка и количество разбиений отрезка.

#### Код программы на языке Си++.

```

#pragma once
.....
#pragma endregion
public: double intSinT (double a, double b, double eps) {
    int n2 = 2;
    for (; (1.0 / n2) > eps; n2*=2);
    double pri = (b - a) / n2;
    double ints = Math::Sin(a) + Math::Sin(b);
    //for (double i = a + pri; i < a + n2 * pri - 2 * pri; i += 2 *
pri) {
        for (int i = 1; i < n2;) {
            ints += 4 * Math::Sin(a + (++i) * pri);
            if (i != n2) ints += 2 * Math::Sin(a + (++i) * pri);
        }
    }
}

```

```

        ints *= (pri / 3);
        return ints;
    }
    public: double intSinP (double a, double b, double eps) {
        double pri,S,x;
        int n = (int)(1 / eps);
        int i;
        pri=(b-a)*1.0/n;
        S=0;
        for(i=0;i<n-1;i++)
        {
            x=a+(i-1)*pri+pri/2;
            S=S+Math::Sin(x);
        }
        S=pri*S;
        return S;
    }
    private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {
        Form1::Close();
    }
    private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
        double a = Convert::ToDouble(textBox1->Text);
        double b = Convert::ToDouble(textBox2->Text);
        double eps = Convert::ToDouble(textBox3->Text);

        double ints = intSinT(a, b, eps);
        textBox4->Text = Convert::ToString(ints);

        ints = intSinP(a, b, eps);
        textBox5->Text = Convert::ToString(ints);
    }
};
}

```

**Визуализация работы программы.**

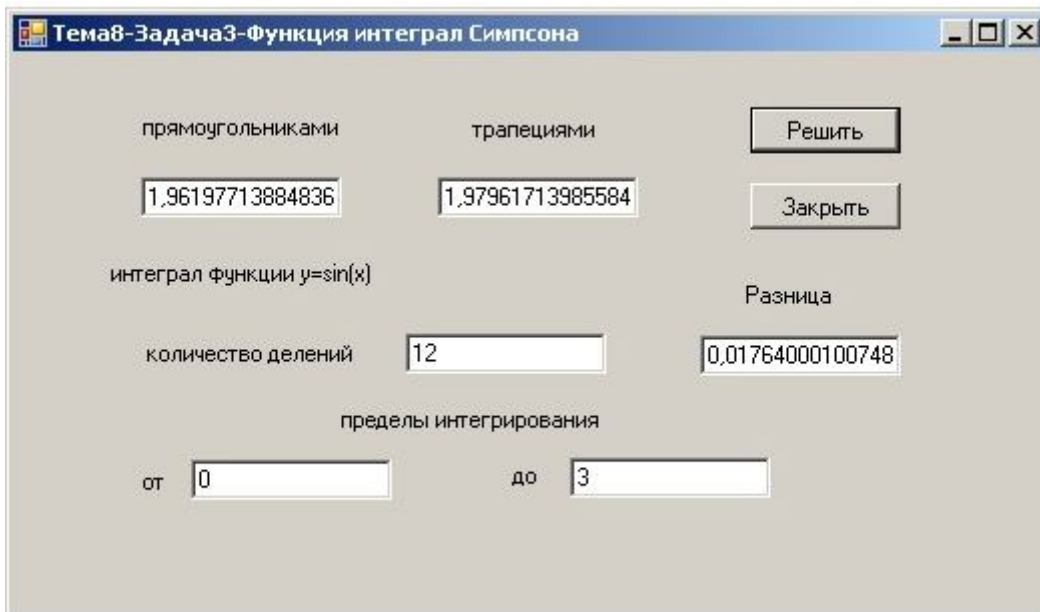


Рис. 31. Визуализация работы программы вычисления функции интеграла методом Симпсона.

#### Задача4. Функция нахождения корня уравнения методом Ньютона.

Найти корень уравнения  $x^2 - y = 0$  по итерационной формуле Ньютона и точностью eps. Написать функцию с одним параметром – положительным вещественным числом y. Результат – корень уравнения, то есть число x.

#### Код программы на языке Си#.

```
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 Тема_8.Задача_4
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void button2_Click(object sender, EventArgs e)
        {
            Form1.ActiveForm.Close();
        }
        unsafe double sqrt(double a, double eps)
        {
            double pribl = 1; // приближение числа a
            double pribl1 = 0;
```

```

for(;;)
{
    pribl1 = 0.5 * (pribl + a / pribl);
    if (Math.Abs(pribl - pribl1) < eps)
        break;
    pribl = pribl1;
}
return pribl;
}
private void button1_Click(object sender, EventArgs e)
{
    double eps = Convert.ToDouble(textBox3.Text);
    double a = Convert.ToDouble(textBox1.Text);
    textBox2.Text = Convert.ToString(sqrt(a, eps));
}
}
}

```

### Визуализация работы программы.

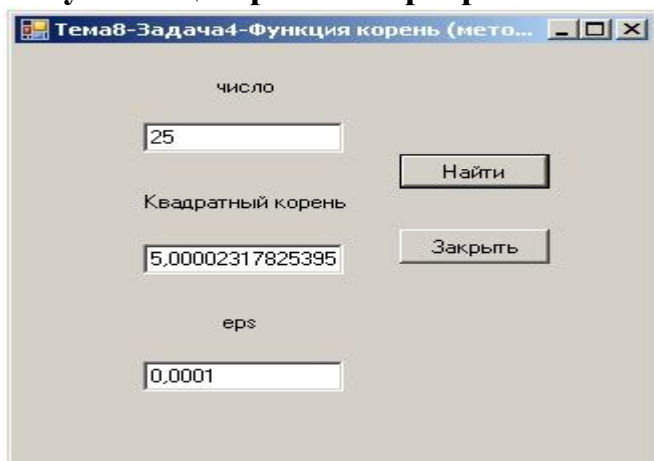


Рис. 32. Визуализация работы программы вычисления функции корня из числа методом Ньютона.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

### Литература по языкам программирования

1. Страуструп, Б. Язык программирования C++: Специальное издание [Текст] / Бьерн Страуструп. – М.: Бином, 2001. – 203 с.
2. Пышкин, Е.В. Основные концепции и механизмы объектно-ориентированного программирования: Учебное пособие [Текст] / Е.В. Пышкин. – СПб.: БХВ-Петербург, 2005. – 130 с.
3. Дейтел, Х. Как программировать на C++ [Текст]: пер. с англ. / Х М. Дейтел – М.: Бином, 2001. – 152 с.
4. Культин, Н.Б. Microsoft Visual C++ в задачах и примерах

[Текст] / Н.Б. Культин – СПб.: БХВ-Петербург, 2010. – 272 с.

5. Романов, Е.Л. Практикум по программированию на С++ [Текст] / Е.Л. Романов – СПб.: БХВ-Петербург, 2004. – 175 с.

6. Климова, Л. С++. Решение типовых задач. [Текст] / Н.Б. Климова Л. – М.: КУДИЦ-ОБРАЗ, 2004. – 191 с.

7. Прата, С. Язык программирования С++: Лекции и упражнения [Текст]: пер. с англ. / Прата С. – СПб.: ООО ДиаСофтЮП, 2004. – 138 с.

8. Шилдт, Г. Самоучитель С++ [Текст]: пер. с англ. / [Текст] / Н.Б. Культин – СПб.: БХВ-Петербург, 2005. – 272 с.

9. Шилдт, Г. С# 4.0: полное руководство [Текст]: пер. с англ. / Герберт Шилдт. – М.: ООО «ИД Вильямс», 2011. - 1056 с.

### **Литература по алгоритмам**

10. Ахо, А. Структуры данных и алгоритмы [Текст]: пер. с англ. / А.Ахо, Д.Хопкрофт, Д.Ульман. – М.: Диалектика, 2003. – 352 с.

11. Долинский, М.С. Алгоритмизация и программирование на Turbo Pascal: от простых до олимпиадных задач [Текст] / М.С. Долинский – СПб.: БХВ-Петербург, 2004. – 82 с.

12. Окулов, С.М. Программирование в алгоритмах [Текст] / С.М. Окулов – М.: Бинوم. Лаборатория знаний, 2004. – 150 с.

13. Долинский, М.С. Решение сложных и олимпиадных задач по программированию: Учебное пособие [Текст] / М.С. Долинский – СПб.: БХВ-Петербург, 2006. – 79 с.

14. Стивен, С. Олимпиадные задачи по программированию: Руководство по подготовке к соревнованиям [Текст]: пер. с англ. / С. Стивен, С. Скиена, А. Мигель, Н. Ревилла. М.: КУДИЦ-ОБРАЗ, 2005. – 119 с.

15. Потопахин, В. Turbo Pascal. Решение сложных задач [Текст] /



Виталий Потопахин – СПб.: БХВ-Петербург, 2006. – 143 с.

16.     Меньшиков, Ф. Олимпиадные задачи по программированию [Текст] / Федор Меньшиков – СПб.: БХВ-Петербург, 2006. – 177 с.

17.     Порублев, И.Н. Алгоритмы и программы. Решение олимпиадных задач. [Текст] / И.Н. Порублев, А.Б. Ставровский. – М.: Диалектика, 2007. – 151 с.

18.     Шень, А. Программирование. Теоремы и задачи. 2 изд. [Текст]: пер. с англ. / А. Шень. – М.: МЦНМО, 2004. – 184 с.