

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение высшего  
образования  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

---

КАФЕДРА №33

ОТЧЕТ ЗАЩИЩЕН С ОЦЕНКОЙ \_\_\_\_\_

ПРЕПОДАВАТЕЛЬ

профессор, кандидат  
технических наук

должность, уч. степень, звание

С.Г.Фомичева

подпись, дата

инициалы, фамилия

**ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 1**

РАЗРАБОТКА КОНСОЛЬНЫХ И WINDOWS FORMS ПРОГРАММ С  
ИСПОЛЬЗОВАНИЕМ ЯЗЫКОВ C#, C++, C  
по курсу: ТЕХНОЛОГИИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ

СТУДЕНТ ГР. № 3032

номер группы

подпись, дата

А.А.Алёшкин

инициалы, фамилия

Санкт-Петербург  
2022

## **Цель работы.**

Получить навыки работы с основными инструментами среды программирования Microsoft Visual Studio 2019. Создать программу с использованием Windows Forms на языках C#, C++, C.

## **Задачи.**

### **Часть 1.**

- 1) Реализовать алгоритм преобразования информации с использованием логических выражений в соответствии с вариантом задания.
- 2) Выполнить проверку на допустимость значений используемых переменных.

### **Вариант №6.**

Реализовать процедуры деления знакового целого числа (вводится с клавиатуры) на число, равное степени 2, с использованием побитовой операции сдвига вправо >>.

### **Часть 2.**

- 1) Создать консольный и Windows Forms проекты, выполняющие поразрядное сложение двух строк (вашей фамилии и имени) с выводом промежуточных результатов.
- 2) Создать программу, выполняющую персональное задание в соответствии с заданным вариантом.

### **Вариант №6.**

1. Дано натуральное число:
  - определить, сколько раз данная цифра A встречается в данном числе (A вводится с клавиатуры);
  - определить, верно ли, что в данном числе сумма цифр больше B, а само число делится на B (B вводится с клавиатуры);
2. Найти все четырёхзначные числа, в которых есть две одинаковые цифры.

## **Часть 1.**

### **Алгоритм решения:**

- Через функцию проверки на допустимость значений используемых переменных вводится два целых числа. Для второго числа также используется проверка на степень двойки;
- Реализовывается функция вычисления степени двойки делителя путём побитового умножения со сдвигаемой в цикле битовой маской;
- В функцию деления передаются делимое и делитель, через вспомогательную функцию вычисляется степень двойки делителя, делимое сдвигается

побитовым сдвигом вправо на вычисленную степень, затем функция возвращает частное от деления.

### Листинг программы:

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <conio.h>

bool if_degree_of_two(register int num)
{
    if (num < 2)
        return false;

    return (num & (num - 1)) == 0;
}

int degree_of_two(register int num)
{
    register int i = 1, tmp = 2;
    while (!(num & tmp))
    {
        i++;
        tmp = tmp << 1;
    }
    return i;
}

int foolproof_int()
{
    int num = 0, input_check = 0, read = 0, char_flag = 0;
    char char_check = 0;
    while (!read) {
        char_flag = 0;
        input_check = 0;
        input_check = scanf("%i", &num);
        char_check = getchar();
        if (char_check == ' ' || char_check == '\t' || char_check == '\n')
            char_flag = 1;
        if (input_check && char_flag) {
            read = 1;
            break;
        }
        printf("Incorrect input. Type again: ");
        scanf("%*s");
    };
    return num;
}

int divide(register int divisible, register int divider)
{
    return divisible >> degree_of_two(divider);
}

int main(void)
{
    register int divisible = 0, divider = 0, result = 0;

    printf("Input divisible:\n");
    divisible = foolproof_int();

    do
    {
        printf("Input divider as a degree of 2:\n");
```

```

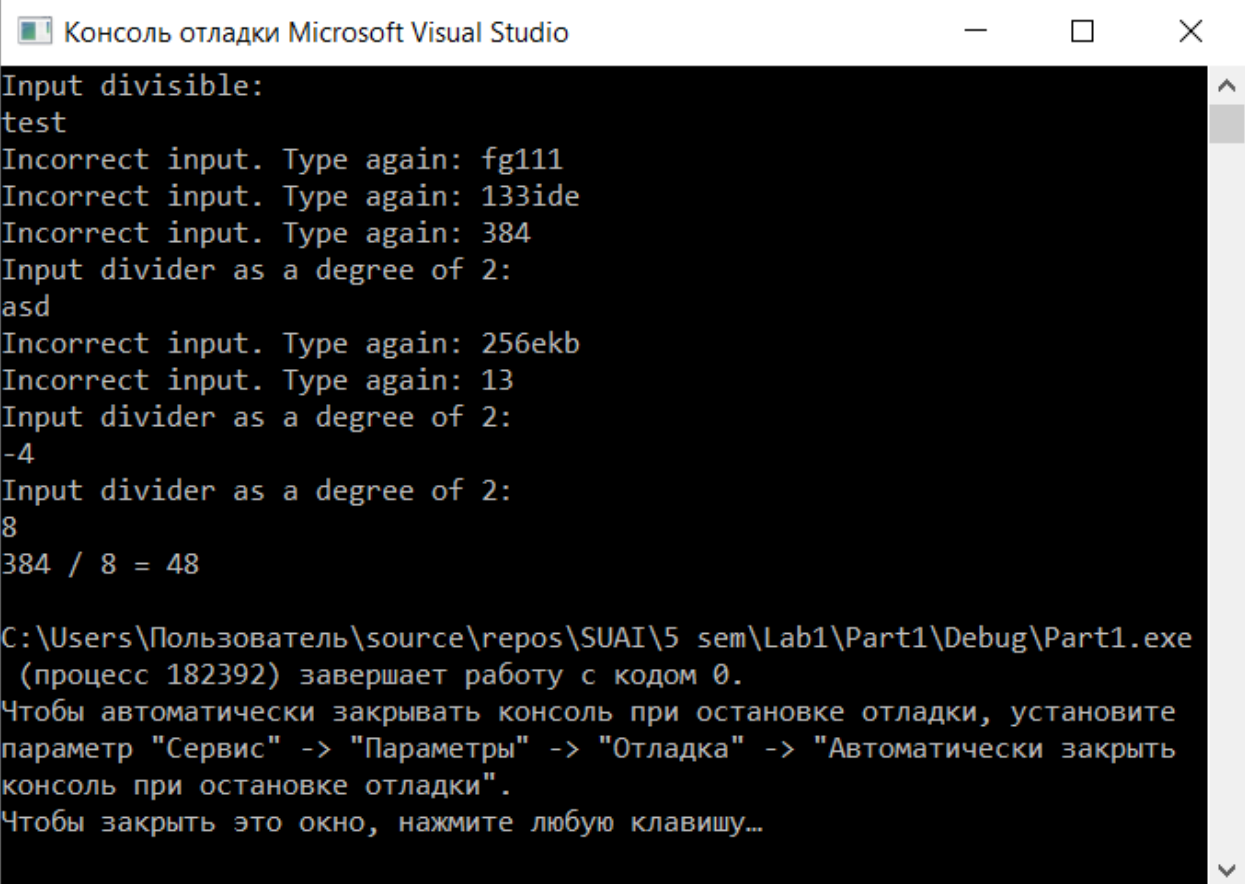
        divider = foolproof_int();
    } while (!if_degree_of_two(divider));

    result = divide(divisible, divider);
    printf("%d / %d = %d \n", divisible, divider, result);

    return 0;
}

```

### Распечатка результатов:



```

Консоль отладки Microsoft Visual Studio
Input divisible:
test
Incorrect input. Type again: fg111
Incorrect input. Type again: 133ide
Incorrect input. Type again: 384
Input divider as a degree of 2:
asd
Incorrect input. Type again: 256ekb
Incorrect input. Type again: 13
Input divider as a degree of 2:
-4
Input divider as a degree of 2:
8
384 / 8 = 48

C:\Users\Пользователь\source\repos\SUAI\5 sem\Lab1\Part1\Debug\Part1.exe
(процесс 182392) завершает работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, установите
параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть
консоль при остановке отладки".
Чтобы закрыть это окно, нажмите любую клавишу...

```

Изображение 1 – Тестирование программы, часть 1.

## Часть 2.1.

### Алгоритм программы:

- Для версии проекта в Windows Forms реализуется интерфейс с двумя текстовыми полями (для ввода имени и фамилии), тремя метками (два для описания текстовых полей и один для вывода результата) и двумя кнопками (для запуска поразрядного сложения строк и для выхода из программы);
- С консоли/текстовых полей считываются две строки: фамилия и имя;
- В ходе программы высчитываются размеры максимальной и минимальной строк;
- По размерам минимальной строки проходит посимвольное сложение строк,

оставшаяся часть дозаписывается в конец результирующей строки;

- С каждой итерацией сложения и дозаписи в результирующую строку промежуточный результат записывается в объект класса `StringBuilder`. По окончании сложения и дозаписи объекта класса `StringBuilder` конвертируется в `String` и выводится в консоль/передаётся в маркировку для вывода результата.

#### Листинг программы консольного проекта:

```
using System;
using System.Text;

namespace Part2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Input your surname:");
            String surname = Console.ReadLine();
            Console.WriteLine("Input your name:");
            String name = Console.ReadLine();

            int resLenght = (surname.Length > name.Length) ? surname.Length :
name.Length;
            int minLenght = (surname.Length > name.Length) ? name.Length :
surname.Length;
            bool ifSurnameLonger = (surname.Length > name.Length) ? true : false;

            StringBuilder res = new StringBuilder();

            int i;
            for (i = 0; i < minLenght; i++)
            {
                Char tmp1 = surname[i];
                Char tmp2 = name[i];
                char tmp = (char)(tmp1 + tmp2);

                res.Append(tmp);
                Console.WriteLine("{0}", res);
            }
            if (ifSurnameLonger)
            {
                for (; i < resLenght; i++)
                {
                    res.Append(surname[i]);
                    Console.WriteLine("{0}", res);
                }
            }
            else
            {
                for (; i < resLenght; i++)
                {
                    res.Append(name[i]);
                    Console.WriteLine("{0}", res);
                }
            }

            Console.WriteLine("Result is {0}", res.ToString());
            Console.ReadKey();
        }
    }
}
```

```

    }
}
}

```

### Листинг программы проекта в Windows Forms:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Part2WinForms
{
    static class Program
    {
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}

```

### Листинг 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 Part2WinForms
{
    public partial class Form1 : Form
    {
        String surname;
        String name;

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }

        private void label1_Click(object sender, EventArgs e)
        {
        }

        private void label2_Click(object sender, EventArgs e)
        {
        }
    }
}

```

```

    }

    private void textBox1_TextChanged(object sender, EventArgs e)
    {
        surname = textBox1.Text;
    }

    private void textBox2_TextChanged(object sender, EventArgs e)
    {
        name = textBox2.Text;
    }

    private void button1_Click(object sender, EventArgs e)
    {
        int resLength = (surname.Length > name.Length) ? surname.Length :
name.Length;
        int minLength = (surname.Length > name.Length) ? name.Length :
surname.Length;
        bool ifSurnameLonger = (surname.Length > name.Length) ? true : false;

        StringBuilder res = new StringBuilder();
        StringBuilder output = new StringBuilder();

        int i;
        for (i = 0; i < minLength; i++)
        {
            Char tmp1 = surname[i];
            Char tmp2 = name[i];
            char tmp = (char)(tmp1 + tmp2);

            res.Append(tmp);
            output.Append(res);
            output.Append('\n');
        }
        if (ifSurnameLonger)
        {
            for (; i < resLength; i++)
            {
                res.Append(surname[i]);
                output.Append(res);
                output.Append('\n');
            }
        }
        else
        {
            for (; i < resLength; i++)
            {
                res.Append(name[i]);
                output.Append(res);
                output.Append('\n');
            }
        }

        lblRes.Text = output.ToString();
    }

    private void lblRes_Click(object sender, EventArgs e)
    {
        lblRes.Text = "Hello World!";
    }

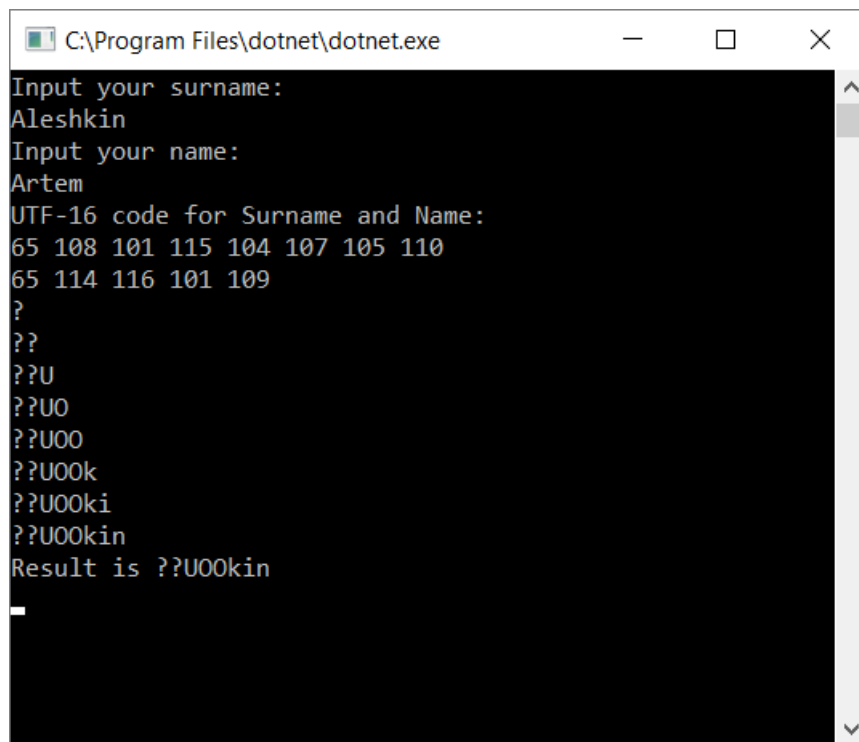
    private void btnClose_Click(object sender, EventArgs e)
    {
        this.Close();
    }

```

} }

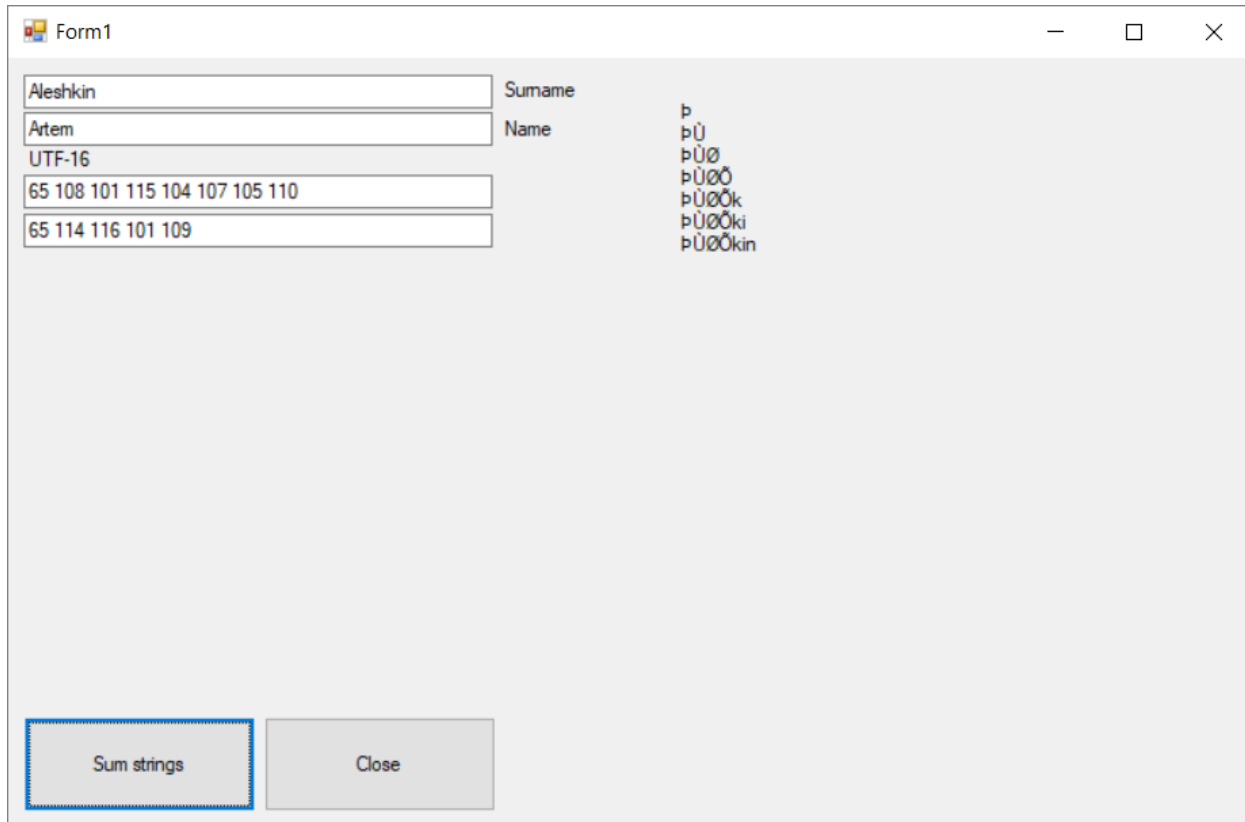


### Распечатка результатов:



Изображение 2 – Тестирование программы, часть 2.1, консольный проект.

**Распечатка изображения форм:**



Изображение 3 – Тестирование программы, часть 2.1, проект Windows Forms.

## Часть 2.2.

### Алгоритм программы:

- Реализовывается вспомогательный метод для проверки на допустимость значений вводимых переменных, считывающий с консоли строку и переводящий её в целое число;
- Реализовывается вспомогательный метод, подсчитывающий количество появлений заданной цифры в заданном числе;
- Реализовывается вспомогательный метод, сравнивающий сумму цифр первого целочисленного аргумента со вторым целочисленным аргументом;
- Реализуется вспомогательный метод, вычисляющий делимость первого целочисленного аргумента на второй;
- Реализуется вспомогательный метод, подсчитывающий и выводящий в консоль все четырёхзначные числа с двумя одинаковыми цифрами;
- В основной части программы считываются входные данные с помощью метода для проверки на допустимость значений вводимых переменных, также для исходного числа стоит дополнительное условие положительности числа (т.к. по условию число натуральное), и для числа A стоит условие диапазона от 0 до 9 (т.к. по условию число A является цифрой);
- С помощью описанных ранее вспомогательных методов вычисляются и выводятся в консоль количество цифр A в исходном числе, сравнение суммы цифр исходного числа с числом B, делимость исходного числа на B а также все четырёхзначные числа с двумя одинаковыми цифрами и их количество.

### Листинг консольного проекта:

```
using System;
using System.Linq;

namespace Part2PersonalTask
{
    class Program
    {
        public static int FoolprofInputInt()
        {
            bool flag = false;
            int number = 0;
            string num;
            while (!flag)
            {
                Console.WriteLine("Input number:");
                num = Console.ReadLine();
                try
                {
                    number = Convert.ToInt32(num);
                }
            }
        }
    }
}
```

```

        flag = true;
    }
    catch (OverflowException)
    {
        Console.WriteLine("{0} is outside of int range", num);
    }
    catch (FormatException)
    {
        Console.WriteLine("' {0}' cannot be recognized as int", num);
    }
}
return number;
}

public static int AmountOfDigits(int number, int digit)
{
    int count = 0, temp = number;

    do
    {
        if (temp % 10 == digit)
            count++;
        temp /= 10;
    }
    while (temp > 0);

    return count;
}

public static bool IsSumOfDigitsBigger(int number, int numB)
{
    int sum = 0, temp = number;

    do
    {
        sum += temp % 10;
        temp /= 10;
    }
    while (temp > 0);

    return (sum > numB);
}

public static bool IfDividable(int number, int numB)
{
    return (number % numB == 0);
}

public static void AllPairsIn4Digits()
{
    int number = 1000, count = 0;
    int[] temp = new int[4];
    for (; number < 10000; number++)
    {
        temp[0] = number % 10;
        temp[1] = number / 10 % 10;
        temp[2] = number / 100 % 10;
        temp[3] = number / 1000;

        if (temp.Distinct().Count() < 4)
        {
            count++;
            Console.WriteLine("{0}", number);
        }
    }
}

```

```

        Console.WriteLine("The amount of 4-digit numbers with 2 identical digits is
{0}", count);
    }

    static void Main(string[] args)
    {
        int number = 0, numA = 0, numB = 0;

        while (number < 1)
        {
            Console.WriteLine("Base number should be natural.");
            number = Program.FoolprofInputInt();
        }

        do
        {
            Console.WriteLine("Number A should be a single digit.");
            numA = Program.FoolprofInputInt();
        } while ((numA > 9) || (numA < 0));

        Console.WriteLine("Number B.");
        numB = Program.FoolprofInputInt();

        int amountOfA = Program.AmountOfDigits(number, numA);
        Console.WriteLine("Digit {0} is met in number {1} {2} times.", numA, number,
amountOfA);

        if(Program.IsSumOfDigitsBigger(number, numB))
            Console.WriteLine("Sum of digits in number {0} is bigger than number
{1}.", number, numB);
        else
            Console.WriteLine("Sum of digits in number {0} is less than number {1}.",
number, numB);

        if(Program.IfDividable(number, numB))
            Console.WriteLine("Number {0} can be divided by number {1}.", number,
numB);
        else
            Console.WriteLine("Number {0} cannot be divided by number {1}.", number,
numB);

        Program.AllPairsIn4Digits();

        Console.ReadKey();
    }
}

```

### Листинг программы проекта Windows Forms:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Part2PersonalTaskWinForms
{
    static class Program
    {
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]

```

```

        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}

```

### Листинг 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 Part2PersonalTaskWinForms
{
    public partial class Form1 : Form
    {
        int number = 0;
        int numA = 0;
        int numB = 0;
        bool flag1 = false;
        bool flag2 = false;
        bool flag3 = false;

        public Form1()
        {
            InitializeComponent();
        }

        public static int FoolprofInputInt()
        {
            bool flag = false;
            int number = 0;
            string num;
            while (!flag)
            {
                Console.WriteLine("Input number:");
                num = Console.ReadLine();
                try
                {
                    number = Convert.ToInt32(num);
                    flag = true;
                }
                catch (OverflowException)
                {
                    Console.WriteLine("{0} is outside of int range", num);
                }
                catch (FormatException)
                {
                    Console.WriteLine("{0}' cannot be recognized as int", num);
                }
            }
            return number;
        }

        public static int AmountOfDigits(int number, int digit)

```

```

    {
        int count = 0, temp = number;

        do
        {
            if (temp % 10 == digit)
                count++;
            temp /= 10;
        }
        while (temp > 0);

        return count;
    }

    public static bool IsSumOfDigitsBigger(int number, int numB)
    {
        int sum = 0, temp = number;

        do
        {
            sum += temp % 10;
            temp /= 10;
        }
        while (temp > 0);

        return (sum > numB);
    }

    public static bool IfDividable(int number, int numB)
    {
        return (number % numB == 0);
    }

    public static String AllPairsIn4Digits()
    {
        StringBuilder res = new StringBuilder();
        int number = 1000, count = 0;
        int[] temp = new int[4];
        for (; number < 10000; number++)
        {
            temp[0] = number % 10;
            temp[1] = number / 10 % 10;
            temp[2] = number / 100 % 10;
            temp[3] = number / 1000;

            if (temp.Distinct().Count() < 4)
            {
                count++;
                res.Append(number);
                res.Append('\n');
            }
        }
        res.Append("The amount of 4-digit numbers with 2 identical digits is " +
count);
        return res.ToString();
    }

    private void lblNum_Click(object sender, EventArgs e)
    {
    }

    private void lblNumA_Click(object sender, EventArgs e)
    {

```

```

}

private void lblNumB_Click(object sender, EventArgs e)
{
}

private void lblRes_Click(object sender, EventArgs e)
{
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
    string num;

    richTextBoxRes.Text = "Base number should be natural.";

    num = textBox1.Text;
    try
    {
        number = Convert.ToInt32(num);
        flag1 = true;
    }
    catch (OverflowException)
    {
        richTextBoxRes.Text = num + " is outside of int range\n";
        flag1 = false;
    }
    catch (FormatException)
    {
        richTextBoxRes.Text = num + " cannot be recognized as int\n";
        flag1 = false;
    }

    if (number < 1)
        flag1 = false;
}

private void textBox2_TextChanged(object sender, EventArgs e)
{
    string num;

    richTextBoxRes.Text = "Number A should be a single digit.";

    num = textBox2.Text;
    try
    {
        numA = Convert.ToInt32(num);
        flag2 = true;
    }
    catch (OverflowException)
    {
        richTextBoxRes.Text = num + " is outside of int range\n";
        flag2 = false;
    }
    catch (FormatException)
    {
        richTextBoxRes.Text = num + " cannot be recognized as int\n";
        flag2 = false;
    }
}

```

```

        if ((numA > 9) || (numA < 0))
            flag2 = false;
    }

    private void textBox3_TextChanged(object sender, EventArgs e)
    {
        string num;

        num = textBox3.Text;
        try
        {
            numB = Convert.ToInt32(num);
            flag3 = true;
        }
        catch (OverflowException)
        {
            richTextBoxRes.Text = num + " is outside of int range\n";
            flag3 = false;
        }
        catch (FormatException)
        {
            richTextBoxRes.Text = num + " cannot be recognized as int\n";
            flag3 = false;
        }
    }

    private void btnRun_Click(object sender, EventArgs e)
    {
        if (textBox1.Text == "" || textBox2.Text == "" || textBox3.Text == "")
        {
            richTextBoxRes.Text = "One of the numbers is empty.\n";
        }
        else if (!flag1 || !flag2 || !flag3)
        {
            richTextBoxRes.Text = "One of the numbers is invalid.\n";
            if (!flag1)
                richTextBoxRes.Text += " 1 number is wrong";
            if (!flag2)
                richTextBoxRes.Text += " 2 number is wrong";
            if (!flag3)
                richTextBoxRes.Text += " 3 number is wrong";
        }
        else
        {
            StringBuilder res = new StringBuilder();

            int amountOfA = Form1.AmountOfDigits(number, numA);
            res.Append("Digit " + numA + " is met in number " + number + " " +
amountOfA + " times.\n");

            if (Form1.IsSumOfDigitsBigger(number, numB))
                res.Append("Sum of digits in number " + number + " is bigger than
number " + numB + " .\n");
            else
                res.Append("Sum of digits in number " + number + " is less than
number " + numB + " .\n");

            if (Form1.IfDividable(number, numB))
                res.Append("Number " + number + " can be divided by number " + numB +
".\n");
            else

```



```

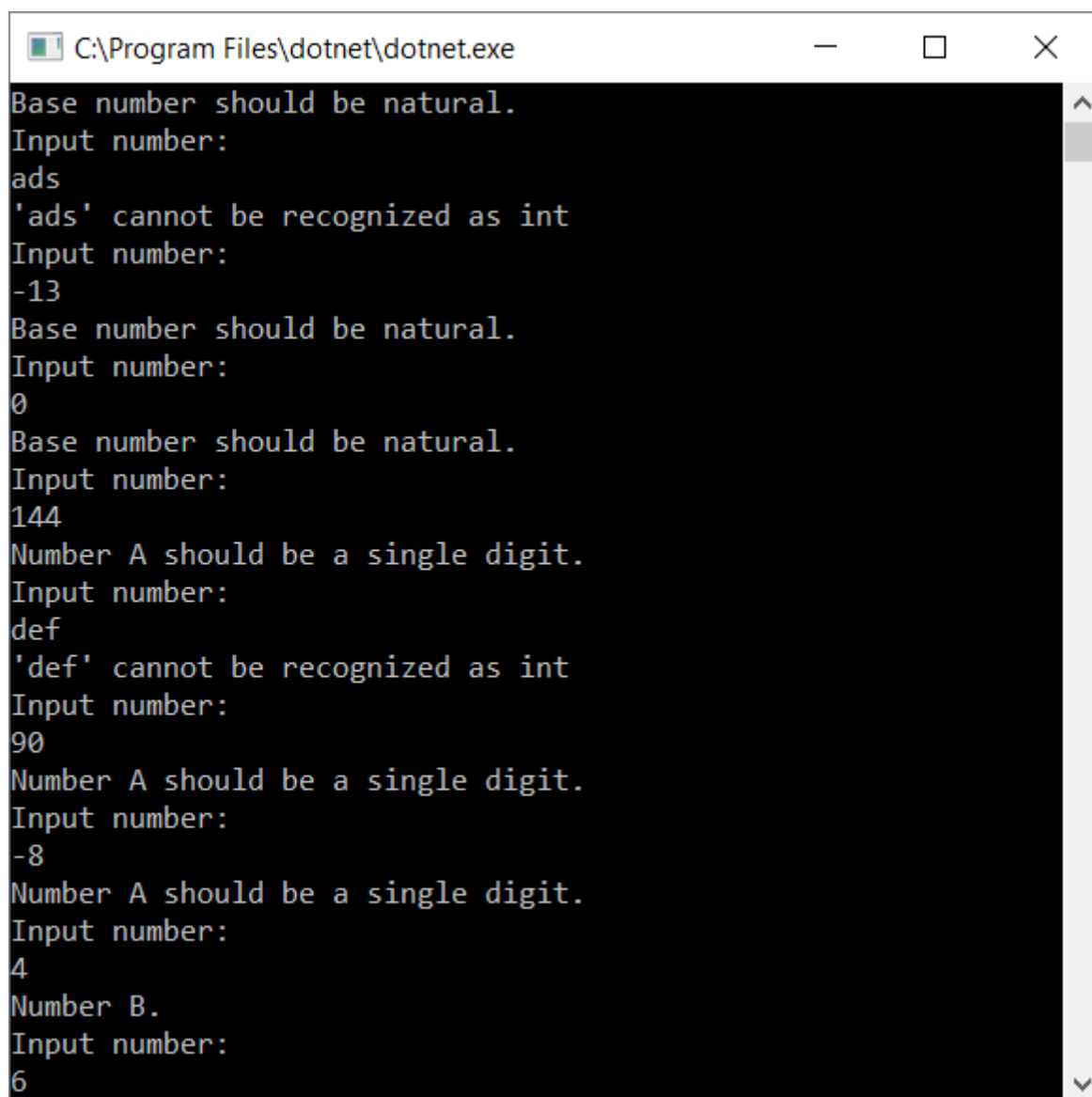
        res.Append("Number " + number + " cannot be divided by number " +
numB + ".\n");
        res.Append(Form1.AllPairsIn4Digits());
        richTextBoxRes.Text = res.ToString();
    }
}

private void btnClose_Click(object sender, EventArgs e)
{
    this.Close();
}

private void richTextBoxRes_TextChanged(object sender, EventArgs e)
{
    richTextBoxRes.Multiline = true;
}
}
}

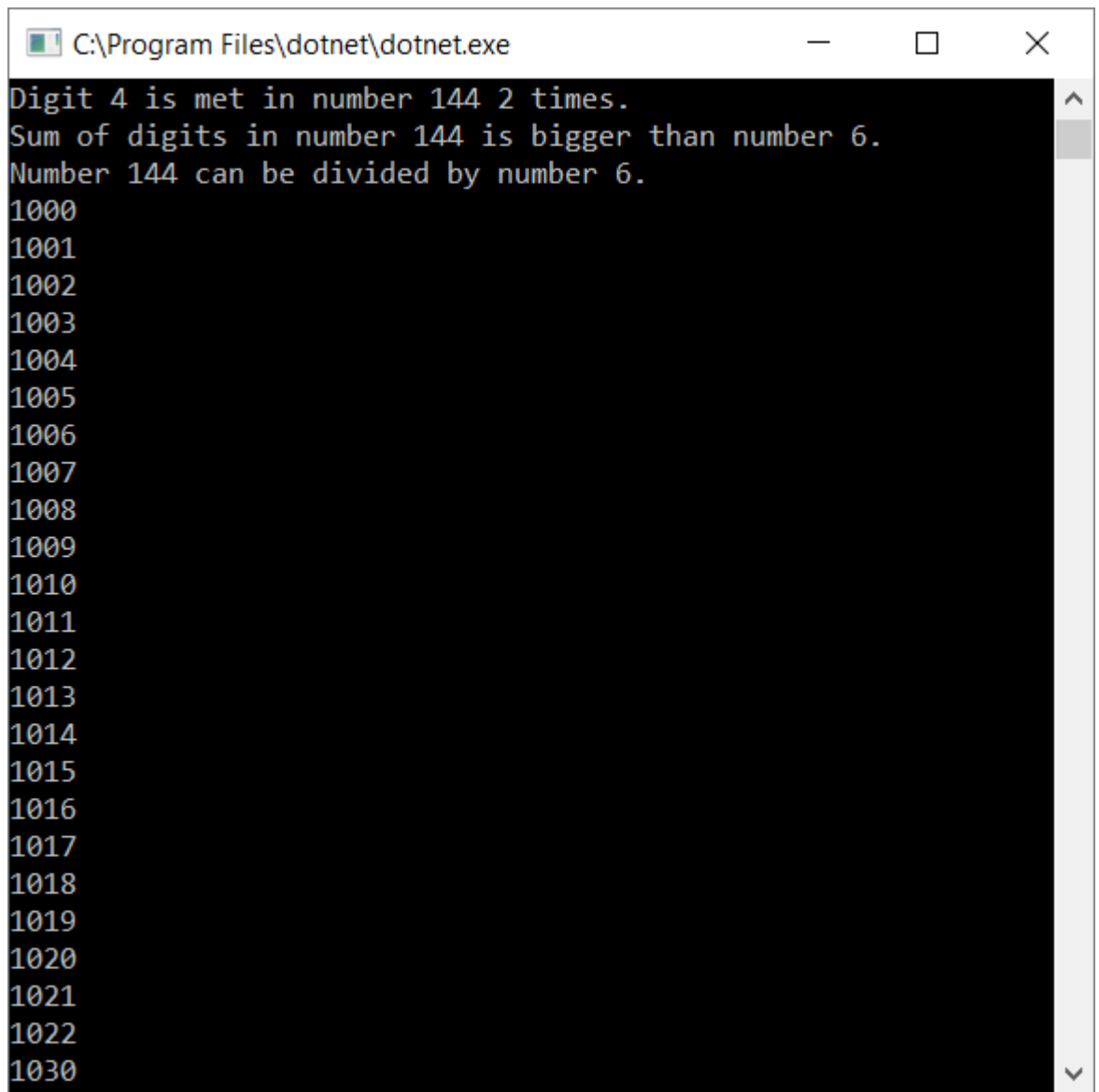
```

### Распечатка результатов:



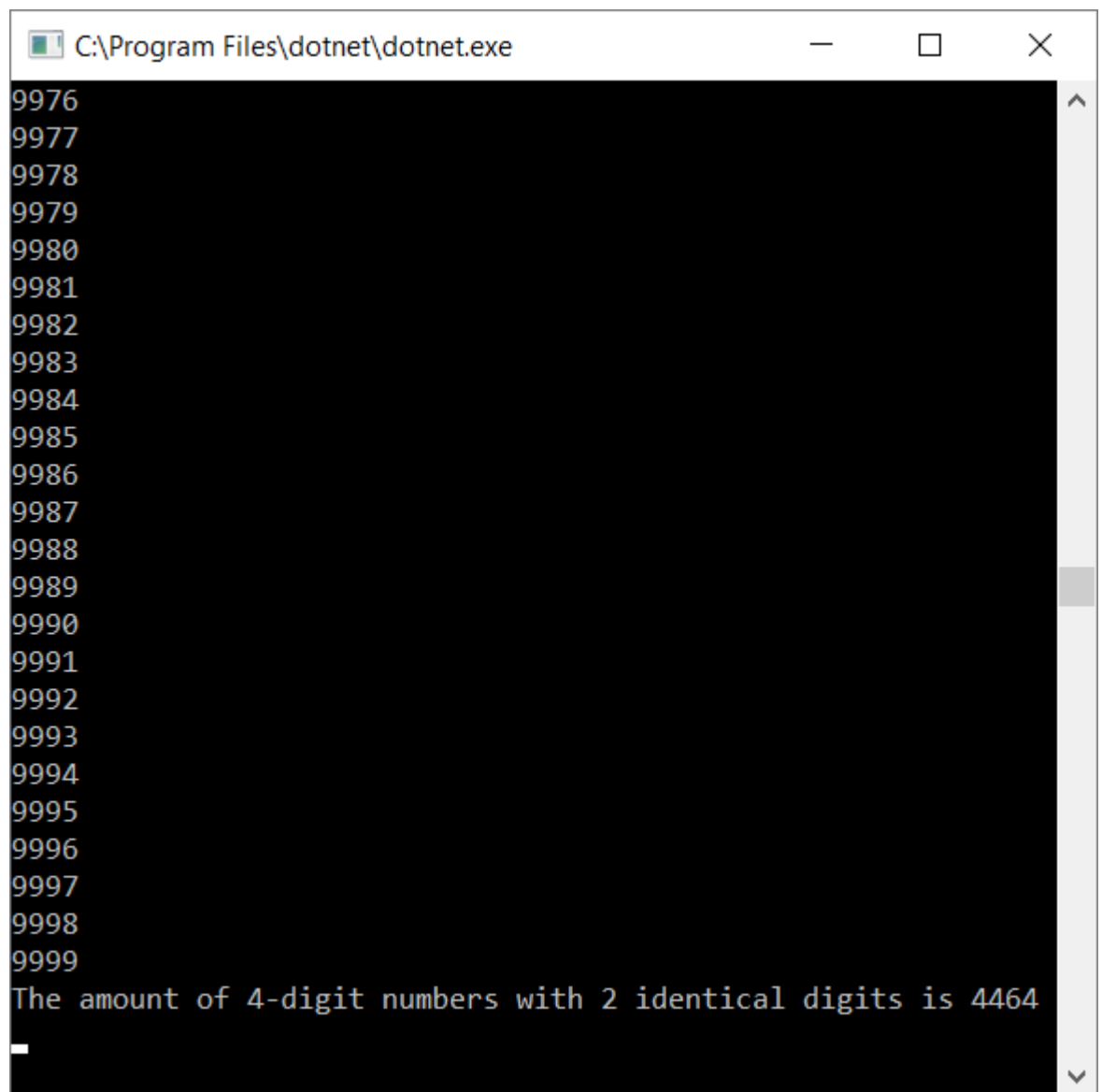
```
C:\Program Files\dotnet\dotnet.exe
Base number should be natural.
Input number:
ads
'ads' cannot be recognized as int
Input number:
-13
Base number should be natural.
Input number:
0
Base number should be natural.
Input number:
144
Number A should be a single digit.
Input number:
def
'def' cannot be recognized as int
Input number:
90
Number A should be a single digit.
Input number:
-8
Number A should be a single digit.
Input number:
4
Number B.
Input number:
6
```

Изображение 4 – Тестирование программы, часть 2.2, проверка на допустимость значений.



```
C:\Program Files\dotnet\dotnet.exe
Digit 4 is met in number 144 2 times.
Sum of digits in number 144 is bigger than number 6.
Number 144 can be divided by number 6.
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1030
```

Изображение 5 – Тестирование программы, часть 2.2, результат выполнения программы.



```
C:\Program Files\dotnet\dotnet.exe
9976
9977
9978
9979
9980
9981
9982
9983
9984
9985
9986
9987
9988
9989
9990
9991
9992
9993
9994
9995
9996
9997
9998
9999
The amount of 4-digit numbers with 2 identical digits is 4464
_
```

Изображение 6 – Тестирование программы, часть 2.2, результат выполнения программы.

## Распечатка изображения форм:

The screenshot shows a Windows form titled "Form1" with standard minimize, maximize, and close buttons in the title bar. The form contains three text input fields on the left, each with a label to its right: "Base number" (containing "-694"), "num A" (containing "99"), and "num B" (containing "14"). Below these fields are two buttons: "Run" and "Close". On the right side of the form is a large text area labeled "Result". The text area contains the following text: "One of the numbers is invalid." followed by "1 number is wrong 2 number is wrong" on the next line.

Изображение 7 – Тестирование форм программы, часть 2.2, проверка на допустимость значений.

The screenshot shows the same Windows form "Form1" after execution. The input fields now contain "144" for "Base number", "4" for "num A", and "6" for "num B". The "Run" button is highlighted with a blue border. The "Result" text area now displays the following output: "Digit 4 is met in number 144 2 times.", "Sum of digits in number 144 is bigger than number 6.", "Number 144 can be divided by number 6.", followed by a list of numbers from 1000 to 1021, each on a new line. A vertical scrollbar is visible on the right side of the text area.

Изображение 8 – Тестирование форм программы, часть 2.2, результат выполнения программы.

Form1

Input	Value
Base number	144
num A	4
num B	6

Run

Close

Result

9976  
9977  
9978  
9979  
9980  
9981  
9982  
9983  
9984  
9985  
9986  
9987  
9988  
9989  
9990  
9991  
9992  
9993  
9994  
9995  
9996  
9997  
9998  
9999  
The amount of 4-digit numbers with 2 identical digits is 4464

Изображение 9 – Тестирование форм программы, часть 2.2, результат выполнения программы.

### Вывод:

В ходе работы были изучены основы работы с графическим интерфейсов Windows Forms и получены навыки программирования на языке C# в среде разработки Microsoft Visual Studio.