

### **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.

### **Задание:**

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 №9.**

Реализовать механизм шифрования и расшифрования символьной информации аддитивным методом, при котором гамма формируется по формуле Фибоначчи.

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 №9.**

1. Дано натуральное число:
  - определить, сколько четных цифр в данном целом числе;
  - определить, верно ли, что в данном числе встречаются цифры А и В (А и В вводятся с клавиатуры).
2. Найти все симметричные четырехзначные числа. Например: 7667, 1331.

### **Часть 1.**

Алгоритм программы:

- С консоли/текстбоксов считывается входная строка и подается на вход шифрующей функции
- Шифрующая функция формирует гамму при помощи генератора Фибоначчи с запаздыванием (lagged Fibonacci generator).
- Дешифрующая функция использует для расшифровки сформированную гамму с заданным смещением относительно начала, равному количеству начальных значений, подаваемых генератору перед шифрованием

Листинг программы:

```
#include <iostream>
#include <vector>

int lfg(int a, int b, std::vector<int>& ks){
    ks.push_back(abs(ks[ks.size() - a] - ks[ks.size() - b]));
    return ks[ks.size() - 1];
}

std::string encrypt(std::string& in, int a, int b, std::vector<int>& ks){
    std::string out = "";
    for(size_t i = 0; i < in.size(); i++)
        out += (char)((in[i] - 'a') ^ (lfg(a, b, ks) % 26)) + 'a';
    return out;
}

std::string decrypt(std::string& in, std::vector<int>& ks, int offset){
    register std::string out = "";
    for(size_t i = 0; i < in.size(); i++)
        out += ((in[i] - 'a') ^ (ks[offset++] % 26)) + 'a';
    return out;
}

int main(){
    register int a = 3, b = 1;
    register std::vector<int> ks = {1, 7, 3, 9};

    std::cout << "Enter string:\t";
    register std::string in;
    std::cin >> in;

    register std::string e = encrypt(in, a, b, ks);
    std::cout << "Encrypt string:\t" << e << "\nDecrypt string:\t" <<
decrypt(e, ks, 4);
}
```

Распечатка результатов:

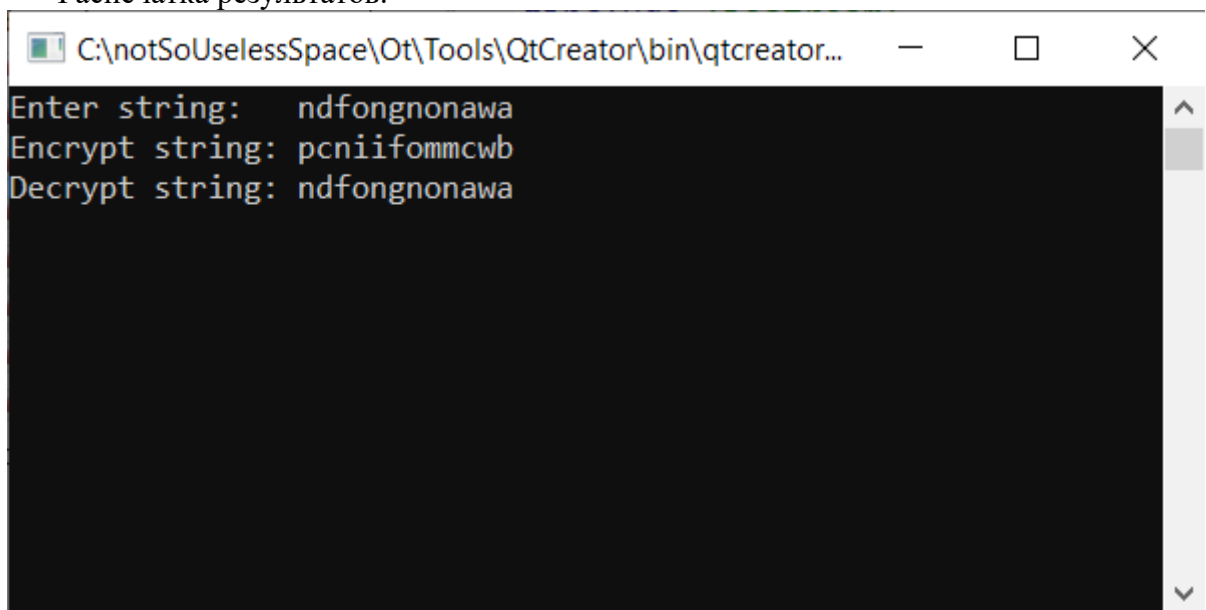
A screenshot of a Qt Creator console window. The title bar shows the path "C:\notSoUselessSpace\Ot\Tools\QtCreator\bin\qtcreator...". The console output is as follows:  
Enter string: ndfongnonawa  
Encrypt string: pcniifommcwb  
Decrypt string: ndfongnonawa

Рисунок 1 – Тестирование программы, часть 1.

## Часть 2.1.

Алгоритм программы:

- С консоли/textbox'ов 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 добавляется в конец результирующей строки;

Листинг программы консольного проекта:

```
using System;
using System.Text;

namespace CSharp_labs {
    class Program {
        public static void p1() {
            string s;
            Console.Write("Enter first string:\t");
            string s1 = Console.ReadLine();
            Console.Write("Enter second string:\t");
            string s2 = Console.ReadLine();

            if(s1.Length > s2.Length) {
                s = s2;
                s2 = s1;
                s1 = s;
            }

            if(s1.Length == 0) {
                Console.Write("Строка 1 не введена\n");
                return;
            }
            if(s2.Length == 0) {
                Console.Write("Строка 2 не введена\n");
                return;
            }

            byte[] b1 = Encoding.Convert(Encoding.UTF8, Encoding.Unicode,
            Encoding.UTF8.GetBytes(s1));
            byte[] b2 = Encoding.Convert(Encoding.UTF8, Encoding.Unicode,
            Encoding.UTF8.GetBytes(s2));

            s = "";
            for(int ii = 0; ii < b1.Length; ii++) s += b1[ii] + " "; s += '\n';
            for(int ii = 0; ii < b2.Length; ii++) s += b2[ii] + " "; s += '\n';

            byte[] b3 = new byte[b2.Length];

            int i = 0;
            for(; i < b1.Length; i++) {
                if(b1[i] == ' ' || b2[i] == ' ') {
                    Console.Write("Обнаружен символ разделения\n");
                    return;
                }
                b3[i] = (byte)((int)b1[i] + (int)b2[i]);
            }
        }
    }
}
```

```

        for(; i < b2.Length; i++) {
            if(b2[i] == ' ') {
                Console.WriteLine("Обнаружен символ разделения\n");
                return;
            }
            b3[i] = b2[i];
        }

        Console.WriteLine(s);
        Console.WriteLine("Answer is:\t\t" + Encoding.Unicode.GetString(b3));
    }

    static void Main(string[] args) {
        p1();
    }
}

```

Листинг программы проекта в Windows Forms:

```
using System;
using System.Text;
using System.Windows.Forms;

namespace WF_labs {
    public partial class L1_p1 : Form {
        public L1_p1() {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e) {
            richTextBox1.Text = "";
            string s = "", s1, s2;
            if(textBox1.Text.Length == 0) {
                MessageBox.Show("Строка 1 не введена\n");
                return;
            }
            if(textBox2.Text.Length == 0) {
                MessageBox.Show("Строка 2 не введена\n");
                return;
            }

            if(textBox1.Text.Length <= textBox2.Text.Length) { s1 = textBox1.Text; s2 =
textBox2.Text; }
            else { s2 = textBox1.Text; s1 = textBox2.Text; }

            byte[] b1 = Encoding.Convert(Encoding.UTF8, Encoding.Unicode,
Encoding.UTF8.GetBytes(s1));
            byte[] b2 = Encoding.Convert(Encoding.UTF8, Encoding.Unicode,
Encoding.UTF8.GetBytes(s2));

            for(int ii = 0; ii < b1.Length; ii++) richTextBox1.Text += b1[ii] + " ";
            richTextBox1.Text += '\n';
            for(int ii = 0; ii < b2.Length; ii++) richTextBox1.Text += b2[ii] + " ";
            richTextBox1.Text += '\n';

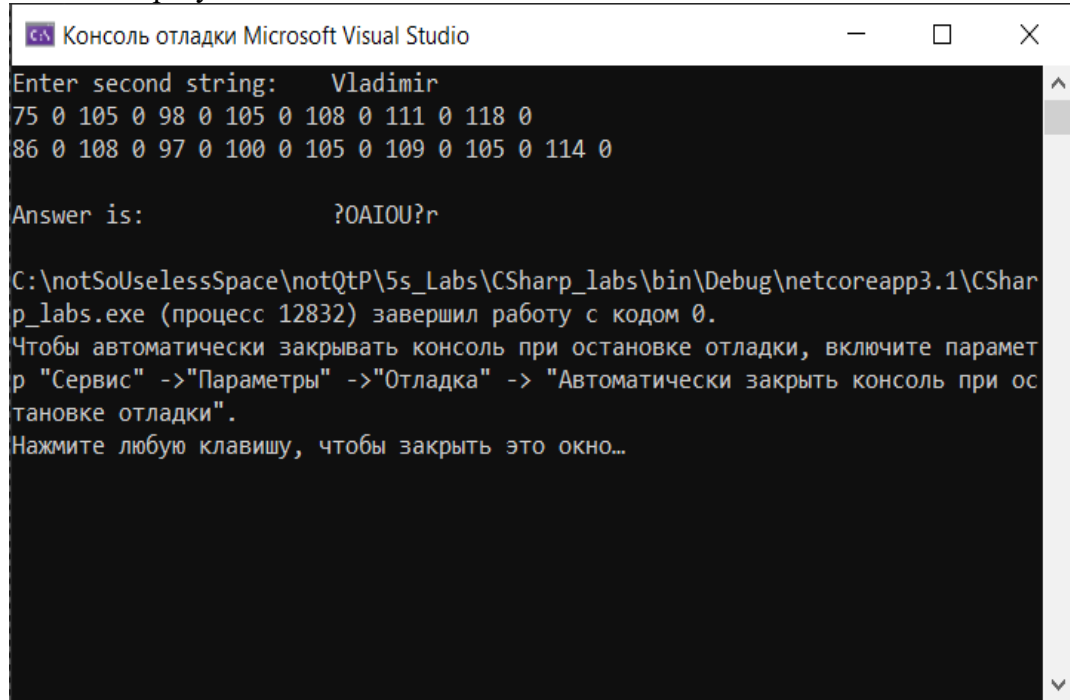
            byte[] b3 = new byte[b2.Length];

            int i = 0;
            for(; i < b1.Length; i++) {
                if(b1[i] == ' ' || b2[i] == ' ') {
                    richTextBox1.Text = "";
                    MessageBox.Show("Обнаружен символ разделения\n");
                    return;
                }
                b3[i] = (byte)((int)b1[i] + (int)b2[i]);
            }

            for(; i < b2.Length; i++) {
                if(b2[i] == ' ') {
                    richTextBox1.Text = "";
                    MessageBox.Show("Обнаружен символ разделения\n");
                    return;
                }
                b3[i] = b2[i];
            }

            richTextBox1.Text += Encoding.Unicode.GetString(b3);
        }
    }
}
```

Распечатка результатов:



Консоль отладки Microsoft Visual Studio

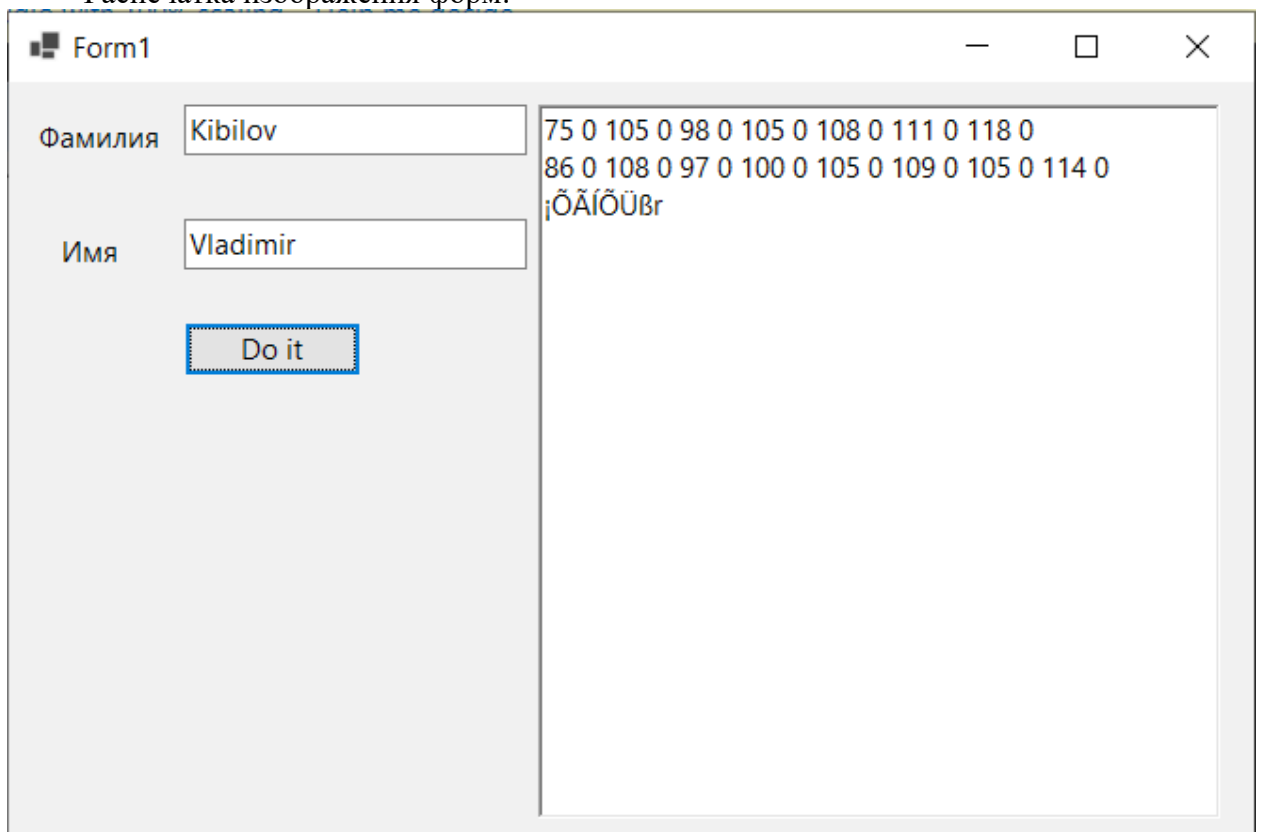
```
Enter second string:  Vladimir
75 0 105 0 98 0 105 0 108 0 111 0 118 0
86 0 108 0 97 0 100 0 105 0 109 0 105 0 114 0

Answer is:             ?0AI0U?r

C:\notSoUselessSpace\notQtP\5s_Labs\CSharp_labs\bin\Debug\netcoreapp3.1\CSharp_labs.exe (процесс 12832) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

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

Распечатка изображения форм:



Form1

Фамилия: Kibilov

Имя: Vladimir

Do it

75 0 105 0 98 0 105 0 108 0 111 0 118 0  
86 0 108 0 97 0 100 0 105 0 109 0 105 0 114 0  
iÖÄfÖÜBr

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

## Часть 2.2.

Алгоритм программы:

- Для первой части программы в поля ввода NumericUpDown вводятся необходимое число и цифры А и В;
- После нажатия на кнопку «Do part 1» программа проходит поразрядно по числу, ведя подсчет четных цифр, и проверяя наличие цифр А и В. после чего выводит результат в поле результата.
- После нажатия на кнопку «Do part 2» программа проходит по числам от 10 до 99 включительно, составляя и выводя симметричные четырехзначные числа путем умножения исходного числа на 100 и добавлением к нему его самого в инвертированном виде, например  $13 * 100 + 31 = 1331$

Листинг Windows Forms:

```
using System;
using System.Text;
using System.Windows.Forms;

namespace WF_labs {
    public partial class L1_p2 : Form {
        public L1_p2() {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e) {
            int cur = (int)numericUpDown1.Value;
            if(cur < 0) cur *= -1;

            int n = 0;
            bool a = false, b = false;
            while(cur > 0) {
                int t = cur % 10;
                if(t % 2 == 0) n++;
                if(t == numberA.Value) a = true;
                if(t == numberB.Value) b = true;
                cur /= 10;
            }

            richTextBox1.Text = string.Format("Количество четных цифр = {0}\nСодержание цифр А и В = {1}\n", n, a & b);
        }

        private void button2_Click(object sender, EventArgs e) {
            StringBuilder sb = new StringBuilder();

            for(int i = 10; i <= 99; i++) {
                sb.Append(i);
                sb.Append(i % 10);
                sb.Append(i / 10 + " ");
            }
            richTextBox1.Text = sb.ToString();
        }
    }
}
```

Распечатка изображения форм:

The screenshot shows a Windows Form titled "L1\_p2". It contains a text box with the value "3242532". Below it are two spin boxes labeled "A" and "B", with values "3" and "2" respectively. There are two buttons: "Do part 1" and "Do part 2". On the right side, there is a text area containing the text: "Количество четных цифр = 4" and "Содержание цифр A и B = True".

Рисунок 4 – Тестирование программы, часть 2.2.1.

The screenshot shows a Windows Form titled "L1\_p2". It contains a text box with the value "0". Below it are two spin boxes labeled "A" and "B", with values "0" and "0" respectively. There are two buttons: "Do part 1" and "Do part 2". On the right side, there is a text area containing a 10x10 grid of numbers, where each number is the sum of its row and column indices (starting from 0). The numbers are: 1001 1111 1221 1331 1441 1551 1661 1771 1881 1991, 2002 2112 2222 2332 2442 2552 2662 2772 2882 2992, 3003 3113 3223 3333 3443 3553 3663 3773 3883 3993, 4004 4114 4224 4334 4444 4554 4664 4774 4884 4994, 5005 5115 5225 5335 5445 5555 5665 5775 5885 5995, 6006 6116 6226 6336 6446 6556 6666 6776 6886 6996, 7007 7117 7227 7337 7447 7557 7667 7777 7887 7997, 8008 8118 8228 8338 8448 8558 8668 8778 8888 8998, 9009 9119 9229 9339 9449 9559 9669 9779 9889 9999.

Рисунок 5 – Тестирование программы, часть 2.2.2.

### Вывод:

Получены навыки работы с основными инструментами среды программирования Microsoft Visual Studio 2019. Созданы программы с использованием Windows Forms на языке C#.