

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

Часть 1.

Задание (Вариант 12*):

- реализовать механизм шифрования и расшифрования битового потока информации, где ключевая последовательность генерируется линейным регистром с обратными связями, определяемыми неприводимым полиномом;
- при описании переменных применить регистровый класс памяти.

Ход работы:

Поток информации, введенный с консоли, записывается в строку, которая далее записывает каждый элемент последовательно в целочисленный массив. Далее в отдельной функции происходит обработка ключа согласно заданному неприводимому полиному, генерируется ключевая последовательность линейным регистром с обратными связями. В функциях шифрования и дешифрования происходит сложение по модулю два ключа с потоком входной информации.

Листинг кода:

```
#include <iostream>

using namespace std;
#include "vector"

int* key_generation(int* key_sequence)
{
    // x^4+x+1 - polynomial
    for (int i = 0; i < 4; i++)
    {
        key_sequence[i] = rand() % (1 - 0 + 1) + 0; // rand() % (b - a + 1) + a - [a,b]
    }
    //key_sequence[4] = {0,1,1,0};
    int feedback = key_sequence[2] ^ key_sequence[3];
    for (int i = 0; i < 2; i++)
    {
        key_sequence[i + 1] = key_sequence[i];
    }
    key_sequence[0] = feedback;

    return key_sequence;
};
```

```

vector<int> encryption(int* key_sequence, vector<int>& input_data)
{
    vector<int> output_data;

    for (int i = 0; i < input_data.size(); i++)
    {
        output_data.push_back(input_data[i] ^ key_sequence[i % 4]);
    }
    return output_data;
}

vector<int> decryption(int* key_sequence, vector<int>& input_data)
{
    vector<int> output_data;

    for (int i = 0; i < input_data.size(); i++)
    {
        output_data.push_back(input_data[i] ^ key_sequence[i % 4]);
    }
    return output_data;
}

int main()
{
    string str = "";
    cout << "Enter the information: ";
    cin >> str;
    vector<int> input_data;
    for (int i = 0; i < str.size(); i++)
    {
        input_data.push_back(str[i] - '0');
    }

    int key[4];

    key_generation(key);

    vector<int> result = encryption(key, input_data);
    cout << "After encryption:" << endl;
    for (int i = 0; i < result.size(); i++)
    {
        cout << result[i] << " ";
    }
    cout << endl;

    result = decryption(key, result);
    cout << "After decryption:" << endl;
    for (int i = 0; i < result.size(); i++)
    {
        cout << result[i] << " ";
    }

    return 0;
}

```

Выполнение программы:

```

Enter the information: 10101011111
After encryption:
0 1 0 0 0 1 0 1 0 0 0
After decryption:
1 0 1 0 1 0 1 1 1 1 1
Process finished with exit code 0

```

Рис.1. Результат выполнения программы.

```

Enter the information: 11111111101
After encryption:
0 0 0 1 0 0 0 1 0 1 0
After decryption:
1 1 1 1 1 1 1 1 1 0 1

```

Рис.2. Результат выполнения программы.

```

Enter the information: 00010101111111110
After encryption:
1 1 1 1 1 0 1 1 0 0 0 1 0 0 0 1 1
After decryption:
0 0 0 1 0 1 0 1 1 1 1 1 1 1 1 1 0

```

Рис.3. Результат выполнения программы.

Часть 2.1.

Задание (Вариант 13):

– создать консольный и Windows Forms проекты, выполняющие поразрядное сложение двух строк (фамилии и имени) с выводом промежуточных результатов.

Часть 2.2.

- найти сумму нечетных делителей натурального числа.
- найти все равновеликие прямоугольники, стороны которых выражены целыми числами A и B , а площадь равна S (A и B принадлежат интервалу от 1 до 20, а S вводится с клавиатуры).

Ход работы:

В первой части задания вводится имя и фамилия, учитывая корректность данных. В зависимости от длины входных строк происходит поразрядное сложение. Оставшаяся часть (та, которая оказалась длиннее) дописывается в конец получившейся строки.

Во второй части задания вводится число, далее в циклах находятся его делители, нечетные из которых суммируются.

В третьей части задания вводится число — площадь заданного прямоугольника. В циклах находятся такие два числа, произведение которых равно введенному числу. Данные числа записываются в линейный список только в том случае, если ранее они не встречались (для избежания повторов). Результат выводится на экран.

Листинг кода консольного проекта:

```
using System.Collections.Specialized;
using System.Data.SqlTypes;
using System.Numerics;
using System.Text;

namespace Program
{
    class Program
    {
        static int Max(int a, int b)
        {
            if (a > b) { return a; }
            return b;
        }

        static int Min(int a, int b)
        {
            if (a > b) { return b; }
            return a;
        }

        static void Main(string[] args)
        {
            Encoding.RegisterProvider(CodePagesEncodingProvider.Instance);
            Encoding win1251 = Encoding.GetEncoding("Windows-1251");

            Console.WriteLine("Enter the name: ");
            string? stringName = Console.ReadLine();
            Console.WriteLine("Enter the surname: ");
            string? stringSurname = Console.ReadLine();
            //Console.WriteLine("{0}", stringName);
            if (stringName.Length == 0 || stringSurname.Length == 0)
            {
                Console.WriteLine("error");
                return;
            }

            byte[] bytesName = win1251.GetBytes(stringName);
            for (int i = 0; i < bytesName.Length; i++)
            {
                Console.Write($"{bytesName[i]} ");
            }

            byte[] bytesSurname = win1251.GetBytes(stringSurname);
            for (int i = 0; i < bytesSurname.Length; i++)
            {
                Console.Write($"{bytesSurname[i]} ");
            }
        }
    }
}
```

```

Console.WriteLine("{0} {1}", length1, length2);

int[] array = new int[Max(stringName.Length, stringSurname.Length)];
//Console.WriteLine("{0}", array.Length);

for (int i = 0; i < (Min(stringName.Length, stringSurname.Length)); i++)
{
    //Console.WriteLine("{0}", stringName[i]);
    array[i] = ((int)stringName[i]) | ((int)stringSurname[i]);
    Console.WriteLine("{0} ", (char)array[i]);
}

for (int i = Min(stringName.Length, stringSurname.Length); i < Max(stringName.Length, stringSurname.Length);
i++)
{
    if (stringName.Length > stringSurname.Length)
    {
        array[i] = ((int)stringName[i]);
        Console.WriteLine("{0} ", (char)array[i]);
    }
    else
    {
        array[i] = ((int)stringSurname[i]);
        Console.WriteLine("{0} ", (char)array[i]);
    }
}

Console.WriteLine();
Console.WriteLine("Enter the number: ");

int a = 0;
bool isNumber = false;

while (!isNumber)
{
    string? stringNumber = Console.ReadLine();
    isNumber = int.TryParse(stringNumber, out a);
    if (!isNumber) Console.WriteLine("Repeat entering the number: ");
}

int k = 0, n = 0;
for (int i = 1; i <= a; i++)
{
    if (a % i == 0)
    {
        if (i % 2 == 1) n += i;
    }
}

Console.WriteLine("The sum of odd divisors is {0}", n);

int s = 0;
bool isNumber = false;
Console.WriteLine("Enter the area of rectangle: ");
while (!isNumber)
{
    string? stringNumber = Console.ReadLine();
    isNumber = int.TryParse(stringNumber, out s);
    if (!isNumber) Console.WriteLine("Repeat entering the number: ");
}

```

```

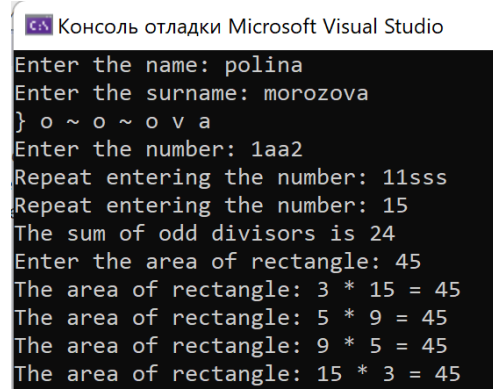
List<int> result = new List<int>();

for (int q = 1; q <= 20; q++)
{
    for (int w = 1; w <= 20; w++)
    {
        if ((q * w) == s)
        {
            if (result.Contains(q) && result.Contains(w))
            {
                break;
            }
            else
            {
                result.Add(q);
                result.Add(w);
            }
        }
    }
}

int j = 0;
for (int i = 0; i < result.Capacity; i++)
{
    Console.WriteLine("The area of rectangle: {0} * {1} = {2}", result[i], result[i + 1], s);
    i++;
}
}
}

```

Выполнение программы:

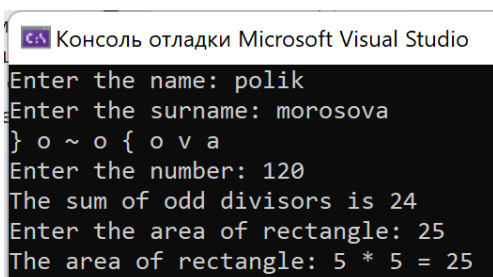


```

Консоль отладки Microsoft Visual Studio
Enter the name: polina
Enter the surname: morozova
} o ~ o ~ o v a
Enter the number: 1aa2
Repeat entering the number: 11sss
Repeat entering the number: 15
The sum of odd divisors is 24
Enter the area of rectangle: 45
The area of rectangle: 3 * 15 = 45
The area of rectangle: 5 * 9 = 45
The area of rectangle: 9 * 5 = 45
The area of rectangle: 15 * 3 = 45

```

Рис.4. Результат выполнения программы.

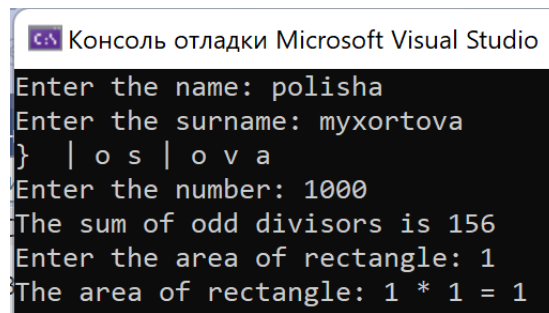


```

Консоль отладки Microsoft Visual Studio
Enter the name: polik
Enter the surname: morosova
} o ~ o { o v a
Enter the number: 120
The sum of odd divisors is 24
Enter the area of rectangle: 25
The area of rectangle: 5 * 5 = 25

```

Рис.5. Результат выполнения программы.



```
Консоль отладки Microsoft Visual Studio
Enter the name: polisha
Enter the surname: myxortova
} | o s | o v a
Enter the number: 1000
The sum of odd divisors is 156
Enter the area of rectangle: 1
The area of rectangle: 1 * 1 = 1
```

Рис.6. Результат выполнения программы.

Листинг кода Windows Forms проекта:

```
using System.Text;

namespace Lab1_1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        static int Max(int a, int b)
        {
            if (a > b) { return a; }
            return b;
        }

        static int Min(int a, int b)
        {
            if (a > b) { return b; }
            return a;
        }

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

        private void button1_Click(object sender, EventArgs e)
        {
            string stringName = textBox1.Text;
            string stringSurname = textBox2.Text;

            string result = "";

            if (stringName.Length == 0 || stringSurname.Length == 0)
            {
                textBox3.Text = "error";
                return;
            }

            int[] array = new int[Max(stringName.Length, stringSurname.Length)];
            //Console.WriteLine("{0}", array.Length);

            for (int i = 0; i < (Min(stringName.Length, stringSurname.Length)); i++)
```

```

    {
        array[i] = ((int)stringName[i]) | ((int)stringSurname[i]);
        result = result + (char)array[i];
        textBox3.Text = result;
        textBox4.Text += array[i] + ", ";
    }

    for (int i = Min(stringName.Length, stringSurname.Length); i < Max(stringName.Length, stringSurname.Length);
i++)
    {
        if (stringName.Length > stringSurname.Length)
        {
            array[i] = ((int)stringName[i]);
        }
        else
        {
            array[i] = ((int)stringSurname[i]);
        }
        result = result + (char)array[i];
        textBox3.Text = result;
        textBox4.Text += array[i] + ", ";
    }

}

private void label1_Click(object sender, EventArgs e)
{

}

private void label2_Click(object sender, EventArgs e)
{

}

private void textBox1_TextChanged(object sender, EventArgs e)
{

}

private void textBox2_TextChanged(object sender, EventArgs e)
{

}

private void textBox3_TextChanged(object sender, EventArgs e)
{

}

private void textBox4_TextChanged(object sender, EventArgs e)
{

}

private void textBox5_TextChanged(object sender, EventArgs e)
{

}

private void label4_Click(object sender, EventArgs e)
{

```



```

}

private void textBox6_TextChanged(object sender, EventArgs e)
{

}

private void button2_Click(object sender, EventArgs e)
{
    int a = 0;
    bool isNumber = false;
    while (!isNumber)
    {
        string? stringNumber = textBox5.Text;
        isNumber = int.TryParse(stringNumber, out a);
        //if (!isNumber) Console.WriteLine("Repeat entering the number: ");
    }

    int k = 0, n = 0;
    for (int i = 1; i <= a; i++)
    {
        if (a % i == 0)
        {
            if (i % 2 == 1) n += i;
        }
    }

    textBox6.Text = n.ToString();
}

private void textBox5_TextChanged_1(object sender, EventArgs e)
{

}

private void label6_Click(object sender, EventArgs e)
{

}

private void label5_Click(object sender, EventArgs e)
{

}

private void textBox8_TextChanged(object sender, EventArgs e)
{

}

private void textBox7_TextChanged(object sender, EventArgs e)
{

}

private void button3_Click(object sender, EventArgs e)
{
    bool isNumber = false;
    int s = 0;
    string res = "";

```

```

while (!isNumber)
{
    string? stringNumber = textBox8.Text;
    isNumber = int.TryParse(stringNumber, out s);
}

List<int> result = new List<int>();
for (int q = 1; q <= 20; q++)
{
    for (int w = 1; w <= 20; w++)
    {
        if ((q * w) == s)
        {
            if (result.Contains(q) && result.Contains(w))
            {
                break;
            }
            else
            {
                result.Add(q);
                result.Add(w);
            }
        }
    }
}

for (int i = 0; i < result.Capacity; i++)
{
    richTextBox1.Text += result[i].ToString() + " * " + result[i + 1].ToString() + " = " + s.ToString() + "\n";
    i++;
}

private void richTextBox1_TextChanged(object sender, EventArgs e)
{
}

private void label7_Click(object sender, EventArgs e)
{
}
}
}

```

Windows forms:

Выполнение программы:

Form1

Enter name: polina

Enter surname: morozova

click

}o~o~ova

Uni:

112, 111, 108, 105, 110, 97,

109, 111, 114, 111, 122, 111, 118, 97,

Enter number: 13

click

The number of odd divisors: 14

Enter number of area: 45

click

Result:

3 * 15 = 45
5 * 9 = 45

Рис.7. Заполненная форма.

Form1

Enter name: polina

Enter surname: polina

click

polina

Uni:

112, 111, 108, 105, 110, 97,

112, 111, 108, 105, 110, 97,

Enter number: 111

click

The number of odd divisors: 152

Enter number of area: 160

click

Result:

8 * 20 = 160
10 * 16 = 160

Рис.8. Заполненная форма.

The screenshot shows a Windows application window titled "Form1". The interface contains the following elements:

- Enter name:** A text box containing "polina".
- Enter surname:** An empty text box.
- click**: A button located below the surname field.
- error**: A text box containing the word "error".
- Uni:** Two empty text boxes stacked vertically.
- Enter number:** A text box containing "21".
- click**: A button located below the number field.
- The number of odd divisors:** A text box containing "32".
- Enter number of area:** A text box containing "100".
- click**: A button located below the area field.
- Result:** A text area containing the following text:
 $5 * 20 = 100$
 $10 * 10 = 100$

Рис.9. Заполненная форма.

Вывод:

Реализован механизм шифрования битового потока информации с ключевой последовательностью, которая генерируется линейным регистром с обратными связями, определяемыми неприводимым полиномом.

Также реализованы консольная и windows Forms программа поразрядного сложения имени и фамилии, поиск числа, являющегося суммой нечетных делителей введенного числа, и поиск неповторяющихся пар чисел, произведение которых составляет площадь – число, введенное с клавиатуры.