

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.

Часть 1.1.

Задание (Вариант 13*):

Реализовать механизм шифрования и расшифрования битового потока информации, где ключевая последовательность генерируется двумя линейными регистрами с обратными связями, определяемыми неприводимым полиномом. Объединение регистров производится операцией XOR (Исключающего ИЛИ)

Ход работы:

Битовый поток информации вводится с клавиатуры. По размеру данной информации формируется неприводимый полином. Генерируется ключевая последовательность линейным регистром с обратной связью, определяемым данным неприводимым полиномом. Генерируется вторая ключевая последовательность линейным регистром с обратной связью, определяемым тем же полиномом, с другими входными данными. Основная ключевая последовательность получается путем объединения двух ключевых последовательностей операцией XOR. В функциях шифрования и дешифрования происходит сложение по модулю 2 основной ключевой последовательности и исходного битового потока информации.

Листинг кода:

```
#include <iostream>
#include <vector>
using namespace std;

//функция формирования полинома
vector<int> polynom(int n) {
    register vector<int> p;
    if (n < 2) {
        p.push_back(0);
        p.push_back(1);
    }
    else if (n < 3 && n >= 2) {
        p.push_back(0);
        p.push_back(1);
    }
    else if (n < 4 && n >= 3) {
```

```

        p.push_back(2);
        p.push_back(1);
        p.push_back(0);
    }
    else if (n < 5 && n >= 4) {
        p.push_back(3);
        p.push_back(1);
        p.push_back(0);
    }
    else if (n < 6 && n >= 5) {
        p.push_back(4);
        p.push_back(1);
        p.push_back(0);
    }
    else if (n < 7 && n >= 6) {
        p.push_back(5);
        p.push_back(2);
        p.push_back(0);
    }
    else if (n < 8 && n >= 7) {
        p.push_back(6);
        p.push_back(1);
        p.push_back(0);
    }
    else{
        p.push_back(7);
        p.push_back(3);
        p.push_back(0);
    }
    return p;
}

//функция генерации ключевой последовательности
int getKey(int data, int size) {
    register unsigned int bits = data;
    register unsigned int key = 0;
    register vector<int> pol = polynom(size);
    for (int i = 0; i < size+1; i++) {
        key = key | (bits << i);
        register int tmp = bits >> 1;
        for (int i = 0; i < pol.size(); i++) {
            bits = bits ^ bits >> pol[i];
        }
        bits <<= pol[0];
        bits |= tmp;
    }
    return key;
}

//функция шифрования
int encryption(int data, int size) {
    register int key = (getKey(data, size) ^ getKey(99, size));
    register int bin = 0;
    for (int i = 0; i < size; i++) {
        bin += pow(2, i);
    }
    return (data ^ key)&bin;
}

//функция генерации основной ключевой последовательности для дешифрования
int enc_key(int data, int size) {
    register int key = (getKey(data, size) ^ getKey(99, size));
    return key;
}

```

```

//функция дешифрования
string decryption(int data, int size, int key) {
    register int bin = 0;
    for (int i = 0; i < size; i++) {
        bin += pow(2, i);
    }
    register int encoding_data = (data ^ key)&bin;
    string sdata = "";
    while (encoding_data != 0) {
        sdata += ((char)(encoding_data % 2) + 48);
        encoding_data /= 2;
    }
    reverse(sdata.begin(), sdata.end());

    return sdata;
}

int main() {
    register string data;
    std::cout << "enter data:\n";
    cin >> data;
    register int size = data.size();
    register int datai = 0;
    for (int i = 0; i < data.size(); i++) {
        if (data[i] == '1')
            datai += std::pow(2, data.size() - 1 - i);
        else if (data[i] != '0') {
            std::cout << "error";
            exit(0);
        }
    }

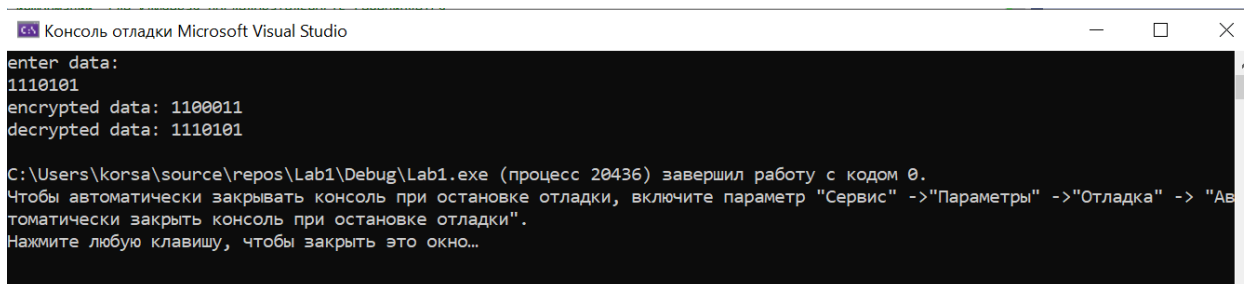
    register int code = encryption(datai, size);
    register int code2 = code;

    register string scode = "";
    while (code != 0) {
        scode += ((char)(code % 2) + 48);
        code /= 2;
    }
    if (scode.size() < size) {
        register string ss = "";
        for (int i = 0; i <= scode.size(); i++) {
            ss += "0";
        }
        for (int i = size - 1; i > scode.size(); i--) {
            ss += (char)scode[i - scode.size() - 1];
        }
        scode = ss;
    }
    std::cout << "encrypted data: " << scode << endl;

    register int k = enc_key(datai, size);
    cout << "decrypted data: " << decryption(code2, size, k) << endl;
}

```

Результаты работы программы:

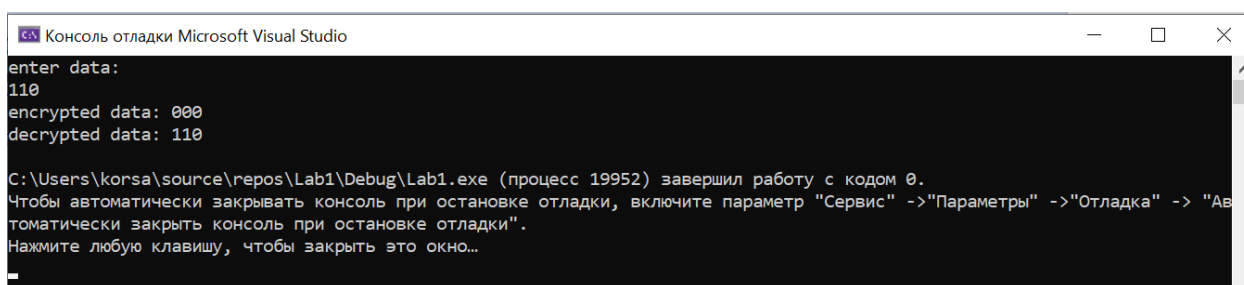


```
Консоль отладки Microsoft Visual Studio

enter data:
1110101
encrypted data: 1100011
decrypted data: 1110101

C:\Users\korsa\source\repos\Lab1\Debug\Lab1.exe (процесс 20436) 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 1 - Результат шифрования и дешифрования введенных данных 1110101

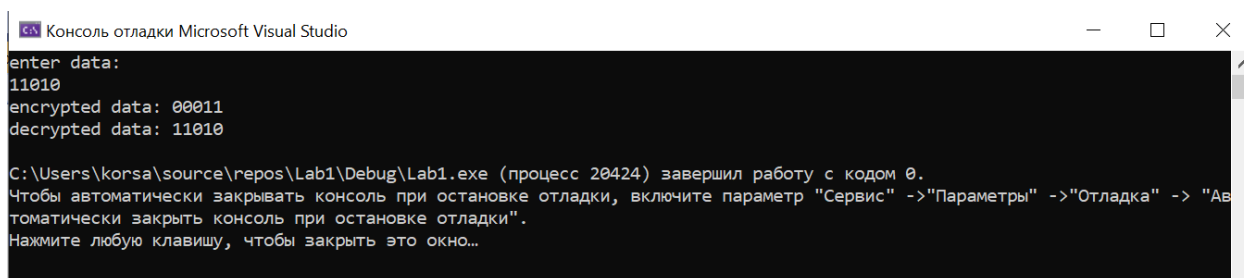


```
Консоль отладки Microsoft Visual Studio

enter data:
110
encrypted data: 000
decrypted data: 110

C:\Users\korsa\source\repos\Lab1\Debug\Lab1.exe (процесс 19952) 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 - Результат шифрования и дешифрования введенных данных 110



```
Консоль отладки Microsoft Visual Studio

enter data:
11010
encrypted data: 00011
decrypted data: 11010

C:\Users\korsa\source\repos\Lab1\Debug\Lab1.exe (процесс 20424) 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 3 - Результат шифрования и дешифрования введенных данных 11010

Часть 2.1.

Задание:

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

1) Консольный проект

Ход работы:

После запуска программы запрашивается ввод имени и фамилии с клавиатуры. Имя и фамилия записываются в переменные типа string. Создаются два массива типа int для имени и фамилии и заполняются кодами ASCII для каждой буквы. Далее производится поразрядное сложение кодов и запись в результирующий массив, размер которого равен размеру большей из введенных строк. Символы, соответствующие данным кодам из результирующего массива, выводятся на экран.

Листинг кода:

```
using System;

namespace name_surname {
    class Program {
        static void Main(string[] args)
        {
            string name = "";
            string surname = "";
            Console.WriteLine("Enter name:");
            name = Console.ReadLine();
            Console.WriteLine("Enter surname:");
            surname = Console.ReadLine();
            int[] n = new int[name.Length];
            int[] s = new int[surname.Length];
            for (int i = 0; i < n.Length; i++) {
                n[i] = (int)(name[i]);
            }
            for (int i = 0; i < s.Length; i++) {
                s[i] = (int)(surname[i]);
            }
            int maxSize = name.Length > surname.Length ? name.Length : surname.Length;
            int[] result = new int[maxSize];
            if (surname.Length > name.Length)
            {
                for (int i = 0; i < result.Length; i++)
                {
                    if (i < n.Length)
                    {
                        result[i] = (n[i] | s[i]);
                    }
                    else
                        result[i] = s[i];
                }
            }
            else {
                for (int i = 0; i < result.Length; i++)
                {
                    if (i < s.Length)
                    {
                        result[i] = (n[i] | s[i]);
                    }
                    else
                        result[i] = n[i];
                }
            }
            Console.WriteLine("Result: ");
            foreach (int i in result)
```

```

        Console.Write((char)(i));
        Console.WriteLine();
    }
}

```

Результат работы программы:

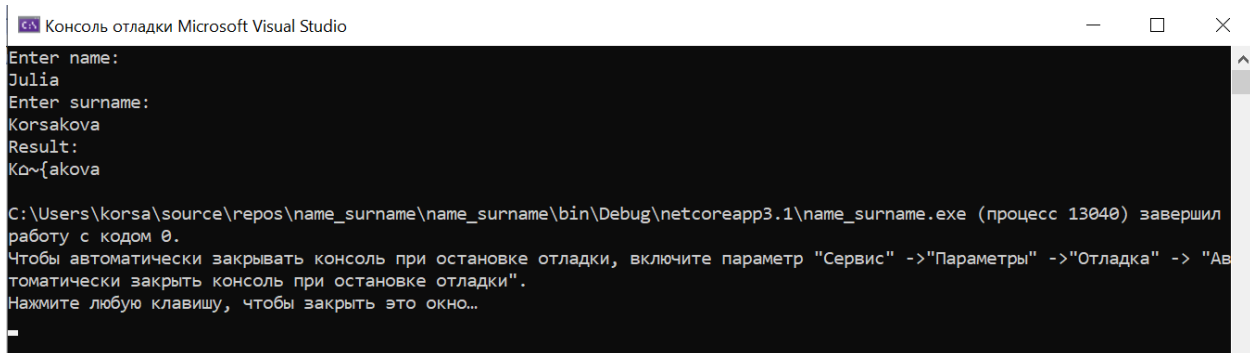


Рисунок 4 - Результат поразрядного сложения строк

2) Windows Forms

Ход работы:

После запуска программы на экране появляется форма, в которой предлагается ввести имя и фамилию в соответствующие текстовые поля. Введенные строки записываются в переменные типа `string`. Проверяется корректность введенных данных. Если в поле для ввода имени или фамилии ввести символ, не являющийся буквой, или оставить поле для ввода пустым, то на экране появится сообщение об ошибке с просьбой ввести данные заново. При вводе данных, в текстовые поля «name code» или «surname code» автоматически выводятся коды ASCII введенных букв. При нажатии на кнопку «result» в текстовом поле «result» появляется результат поразрядного сложения двух строк, в текстовое поле «result code» выводится код поразрядного сложения двух строк.

Листинг кода:

```

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

namespace winf_name_surname
{
    public partial class Form1 : Form
    {
        string name;
        string surname;
    }
}

```

```

public Form1()
{
    InitializeComponent();
    buttonOk.Click += button_click;
}

private void textName_TextChanged(object sender, EventArgs e) {
    name = textName.Text;
    for (int i = 0; i < name.Length; i++)
    {
        if (name[i] < 65 || name[i] > 122) {
            textName.Clear();
            textCodeN.Clear();
            MessageBox.Show("enter again");
        }
    }
    byte[] asciiBytesName = Encoding.ASCII.GetBytes(name);

    string textname = System.Text.Encoding.ASCII.GetString(asciiBytesName);
    textCodeN.Text = String.Join(" ", Array.ConvertAll(asciiBytesName, byteValue =>
byteValue.ToString()));
}

private void texSurname_TextChanged(object sender, EventArgs e) {
    surname = textSurname.Text;
    for (int i = 0; i < surname.Length; i++)
    {
        if (surname[i] < 65 || surname[i] > 122)
        {
            textSurname.Clear();
            textCodeS.Clear();
            MessageBox.Show("enter again");
        }
    }
    byte[] asciiBytesSurname = Encoding.ASCII.GetBytes(surname);

    string textsurname = System.Text.Encoding.ASCII.GetString(asciiBytesSurname);
    textCodeS.Text = String.Join(" ", Array.ConvertAll(asciiBytesSurname, byteValue =>
byteValue.ToString()));
}

//функция поразрядного сложения двух строк
private string BitAddition(string n, string s) {
    int[] result;
    if (n.Length > s.Length) {
        result = new int[n.Length];
        for (int i = 0; i < s.Length; i++) {
            result[i] = ((int)(n[i])) | ((int)(s[i]));
        }
        for (int i = s.Length; i < n.Length; i++) {
            result[i] = (int)n[i];
        }
    }
    else {
        result = new int[s.Length];
        for (int i = 0; i < n.Length; i++) {
            result[i] = ((int)(n[i])) | ((int)(s[i]));
        }
        for (int i = n.Length; i < s.Length; i++) {
            result[i] = (int)s[i];
        }
    }
    string str = "";
    for (int i = 0; i < result.Length; i++) {
        str += (char)result[i];
    }
}

```

```

    }
    return str;
}
private void button_click(object sender, EventArgs e) {
    if (textName.TextLength == 0 || textSurname.TextLength == 0) {
        MessageBox.Show("error! enter all data");
        return;
    }
    string result = BitAddition(name, surname);
    byte[] asciiBytesRes = Encoding.ASCII.GetBytes(result);

    string resBytes = System.Text.Encoding.ASCII.GetString(asciiBytesRes);
    textResultCode.Text = String.Join(" ", Array.ConvertAll(asciiBytesRes, byteValue =>
byteValue.ToString()));
    textResult.AppendText(result);
}
}
}

```

Результаты работы программы:

Рисунок 5 - Пустая форма

The screenshot shows a Windows application window titled 'Form1'. It contains several text input fields and a button. On the left, there are two input fields: the top one contains 'Julia' and the bottom one contains 'Korsakova'. In the center, there is a button labeled 'result' with a blue border. On the right, there are three input fields: the top one is labeled 'name code' and contains '74 117 108 105 97'; the middle one is labeled 'surname code' and contains '75 111 114 115 97 107 111 118 97'; the bottom one is labeled 'result code' and contains '75 127 126 123 97 107 111 118 97'. At the bottom left, there is an input field labeled 'result' containing 'K~{akova'.

Рисунок 6 - Заполненная форма

Часть 2.2.

Задание: (Вариант 7)

1. Дано натуральное число:

- Найти вторую(слева) цифру данного числа
- Определить, верно ли что данное число делится на А, В и С (А, В и С вводятся с клавиатуры)

2. Найти все двухзначные числа, которые при умножении на 2 заканчиваются на 8, а при умножении на 3 - на 4.

Ход работы:

1. Найти вторую слева цифру данного числа

Вводится число. Проверяется корректность ввода данных. Если в соответствующее текстовое окно вводится не цифра, то программа выдает сообщение об ошибке с просьбой ввести число заново. Если введенное число одноразрядное, то функция возвращает -1 и программа выводит сообщение, что в данном числе только одна цифра. Если число двухразрядное, то функция возвращает данное число по модулю 10. Если в числе более 2 разрядов, то в функции происходит деление числа на 10, пока число не станет двухразрядным, после чего возвращает значение равное

остатку от деления на 10 этого числа. Вторая цифра числа выводится в соответствующее текстовое поле после нажатия на кнопку «get 2nd digit of this number»

2. Определить, верно ли что данное число делится на A, B и C

Числа A, B и C вводятся в соответствующие текстовые поля формы. Производится проверка корректности введенных данных. Если пользователь вводит не цифры или оставляет соответствующие текстовые поля пустыми, то программа выдает сообщение об ошибке с просьбой ввести данные заново. Функция вычисляет остаток от деления введенного числа на A, B и C, и если все остатки равны нулю, то возвращает true, иначе false. При нажатии на кнопку «Is the number divided by A B and C?» появляется сообщение, в котором написано делится данное число на все 3 числа или нет.

3. Найти все двухзначные числа, которые при умножении на 2 заканчиваются на 8 или при умножении на 3 - на 4.

В функции в цикле от 10 до 99 включительно числа умножаются на 2 и берется остаток по модулю 10. Если данный остаток равен 8, то число записывается в одну строку. В этом же цикле число умножается на 3 и берется остаток по модулю 10. Если этот остаток равен 4, то число записывается в другую строку. При нажатии на кнопку, появляется сообщение с двумя строками. В строке «2:..» написаны все двухзначные числа, которые при умножении на 2 заканчиваются на 8. В строке «3:..» написаны все двухзначные числа, которые при умножении на 3 заканчиваются на 4.

Листинг кода:

```
using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace Lab1Var7Form
{
    public partial class Form1 : Form
    {
        int num;
        int aa;
        int bb;
        int cc;
        public Form1()
        {
            InitializeComponent();
            buttonGetDigit.Click += buttonClick;
            division.Click += buttonDivisionClick;
            allNums.Click += allNumsClick;
        }
    }
}
```

```

public void number_TextChanged(object sender, EventArgs e) {
    string n = number.Text;
    for (int i = 0; i < n.Length; i++) {
        if (n[i] < 48 || n[i] > 57) {
            number.Clear();
            MessageBox.Show("invalid format");
            return;
        }
    }
    if (n.Length != 0)
    {
        num = Convert.ToInt32(n);
    }
}

public void a_TextChanged(object sender, EventArgs e) {
    string n = a.Text;
    for (int i = 0; i < n.Length; i++)
    {
        if (n[i] < 48 || n[i] > 57)
        {
            a.Clear();
            MessageBox.Show("invalid format");
            return;
        }
    }
    if (n.Length != 0)
    {
        aa = Convert.ToInt32(n);
    }
}

public void b_TextChanged(object sender, EventArgs e)
{
    string n = b.Text;
    for (int i = 0; i < n.Length; i++)
    {
        if (n[i] < 48 || n[i] > 57)
        {
            b.Clear();
            MessageBox.Show("invalid format");
            return;
        }
    }
    if (n.Length != 0)
    {
        bb = Convert.ToInt32(n);
    }
}

public void c_TextChanged(object sender, EventArgs e)
{
    string n = c.Text;
    for (int i = 0; i < n.Length; i++)
    {
        if (n[i] < 48 || n[i] > 57)
        {
            c.Clear();
            MessageBox.Show("invalid format");
            return;
        }
    }
    if (n.Length != 0)

```

```

        {
            cc = Convert.ToInt32(n);
        }
    }

    public void buttonClick(object sender, EventArgs e)
    {
        if (number.TextLength == 0) {
            MessageBox.Show("Error! Enter number");
            return;
        }
        int t = sec_left_num();
        if (t == -1)
            MessageBox.Show("This number has only one digit");
        if (t != -1)
        {
            secDigit.Clear();
            secDigit.AppendText(t+"");
        }
    }

    public void buttonDivisionClick(object sender, EventArgs e)
    {
        if (a.TextLength == 0 || b.TextLength == 0 || c.TextLength == 0 || number.TextLength
== 0)
        {
            MessageBox.Show("Error! Enter all numbers!");
            return;
        }
        if (isDiv())
        {
            MessageBox.Show(num + " is divided by " + aa + ", " + bb + " and " + cc);
        }
        else {
            MessageBox.Show(num + " is not divided");
        }
    }

    public void allNumsClick(object sender, EventArgs e) {
        MessageBox.Show(all_nums());
    }

    int sec_left_num()
    {
        if (num < 10)
            return -1;
        else if (num < 100)
        {
            return (int)(num % 10);
        }
        else
        {
            int t = num;
            int count = 0;
            while (t != 0)
            {
                t /= 10;
                count++;
            }
            t = num / (int)Math.Pow(10, count - 2);
            t = t % 10;
            return t;
        }
    }

```

```

    }

    bool isDiv()
    {
        if (aa == 0 || bb == 0 || cc == 0)
            return false;
        return num % aa == 0 && num % bb == 0 && num % cc == 0;
    }

    string all_nums()
    {
        List<Int32> list3 = new List<int>();
        List<Int32> list2 = new List<int>();
        for (int i = 10; i < 100; i++)
        {
            if ((i * 3) % 10 == 4)
                list3.Add(i);
            if ((i * 2) % 10 == 8)
                list2.Add(i);
        }
        string s = "";
        s += "3: ";
        for (int i = 0; i < list3.Count; i++) {
            s += list3[i] + " ";
        }
        s += "\n2: ";
        for (int i = 0; i < list2.Count; i++)
        {
            s += list2[i] + " ";
        }
        return s;
    }
}

```

Результаты выполнения программы:

The screenshot shows a Windows application window titled "Form1". It contains three distinct sections, each identified by a blue square with a white number in the top-left corner.

- Section 1:** Contains a text input field with the placeholder text "Number". To its right is a button labeled "get 2nd digit of this number:". Further right is an empty rectangular box.
- Section 2:** Contains the text "Enter A, B and C" centered at the top. Below this are three text input fields. The first is labeled "A", the second is labeled "B", and the third is empty. Below these fields is a button labeled "Is the number divisible by A,B and C?".
- Section 3:** Contains a single button with the text "All double-digit x: ((x*3)%10=4)((x*2)%10=8)".

Рисунок 7 - Пустая форма

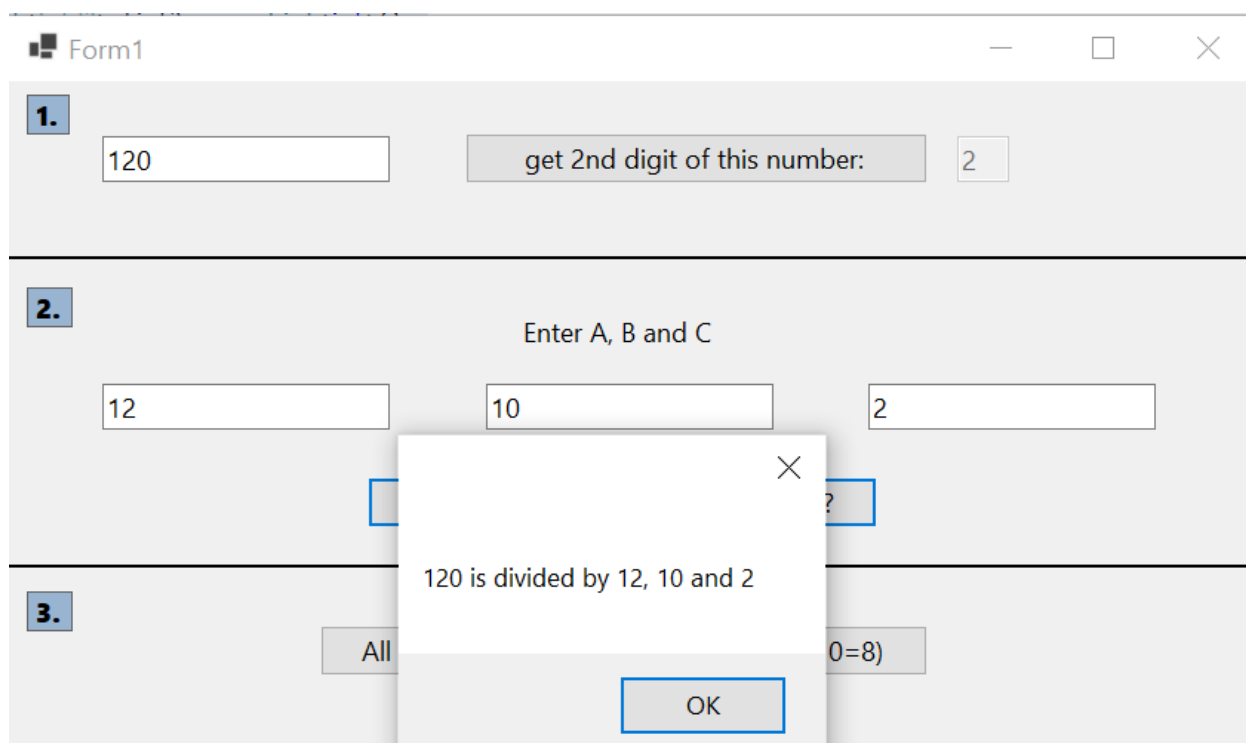


Рисунок 8 - Вторая цифра числа и проверка на деление на A B и C

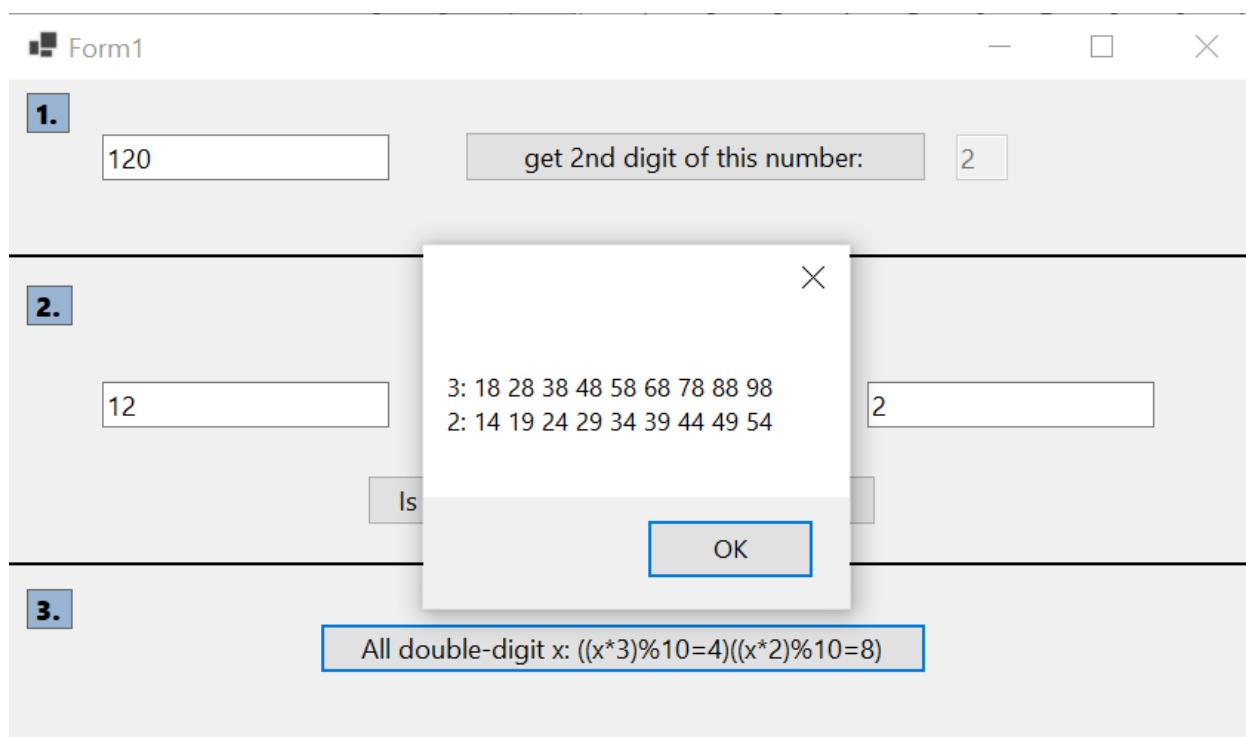


Рисунок 9 - Все двузначные числа, которые при умножении на 2 заканчиваются на 8 или при умножении на 3 заканчиваются на 4

Выводы:

1. Получены навыки работы с основными инструментами среды программирования Microsoft Visual Studio 2019.

2. Создана программа с использованием Windows Forms на языке C# и консольное приложение на языке C#, выполняющие поразрядное сложение строк.

3. Создана программа с использованием Windows Forms на языке C#, выполняющая поиск второй цифры числа, проверку деления числа на 3 заданных числа, поиск всех двузначных чисел, которые при умножении на 2 заканчиваются на 8 или при умножении на 3 заканчиваются на 4.

4. Реализован механизм шифрования и дешифрования битового потока информации, где ключевая последовательность генерируется двумя линейными регистрами с обратными связями, определяемыми неприводимым полиномом на языке C++ с использованием среды программирования Microsoft Visual Studio 2019.