

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет «Информатика и вычислительная техника»

Кафедра «Кибербезопасность информационных систем»

Лабораторная работа №2

по дисциплине: «Методы и средства криптографической защиты информации»

На тему «Шифрованиес использованием систем Цезаря и систем Трисемуса»

Выполнил обучающийся гр. ВКБ42

Михайлов Александр Сергеевич

          Проверила:

Сафарьян О.А

Ростов-на-Дону

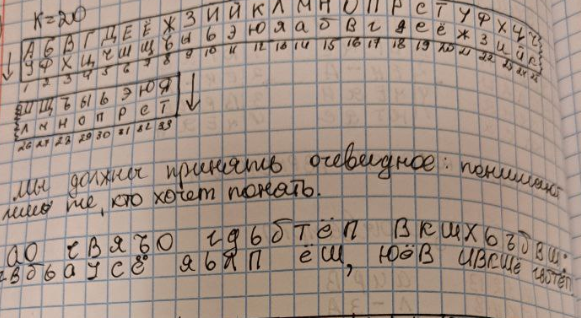
2024

Цель работы: Формирование умений шифрования с использованием систем Цезаря и системы Трисемуса.

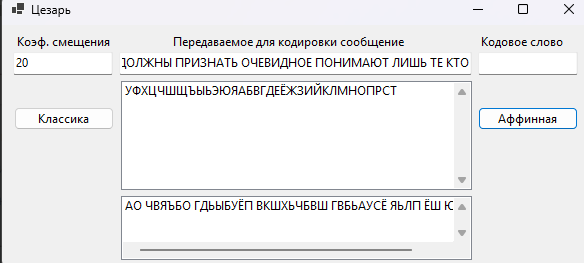
Задание 1 Вариант 20

Зашифруйте сообщение «МЫ ДОЛЖНЫ ПРИЗНАТЬ ОЧЕВИДНОЕ: ПОНИМАЮТ ЛИШЬ ТЕ, КТО ХОЧЕТ ПОНЯТЬ», используя систему Цезаря со значением ключа соответствующим номеру варианта

Ручная отработка:

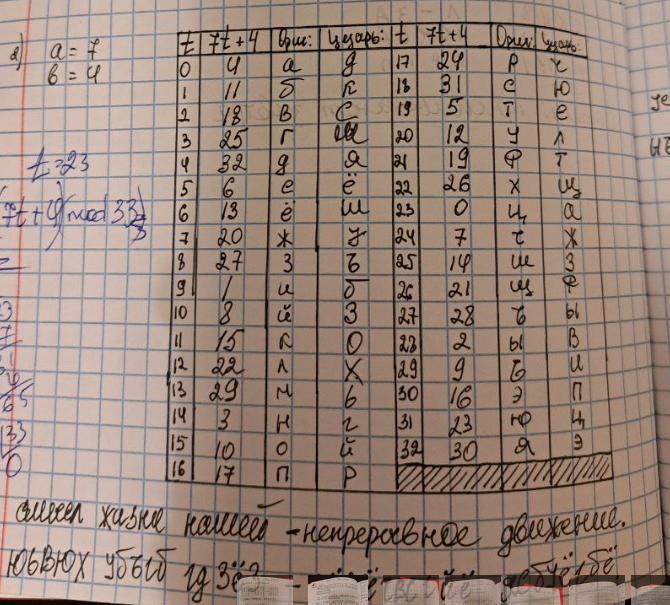


Программная реализация:

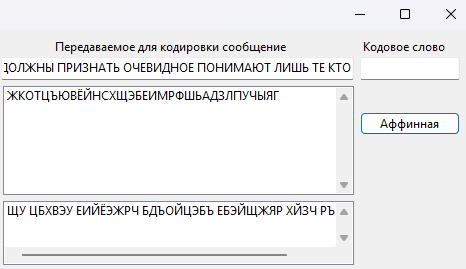


Задание 2

Зашифруйте сообщение «СМЫСЛ ЖИЗНИ НАШЕЙ – НЕПРЕРЫВНОЕ ДВИЖЕНИЕ», используя аффинную систему подстановок Цезаря с ключами, согласно своему варианту. a = 7, b = 4

Ручная реализация:  


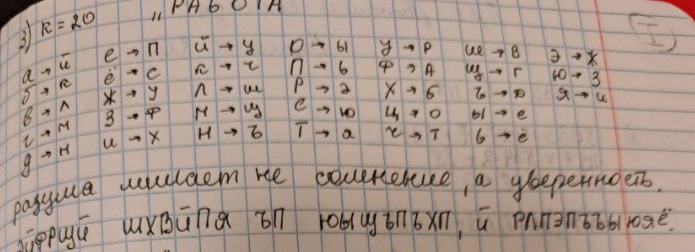
Программная реализация:



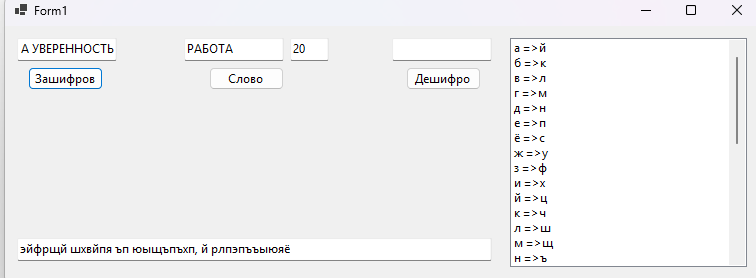
Задание 3

Выполните шифрование сообщения «РАЗУМА ЛИШАЕТ НЕ СОМНЕНИЕ, А УВЕРЕННОСТЬ», используя систему шифрования Цезаря с ключами, соответствующими варианту. k = 20, ключевое слово «РАБОТА».

Ручная реализация:



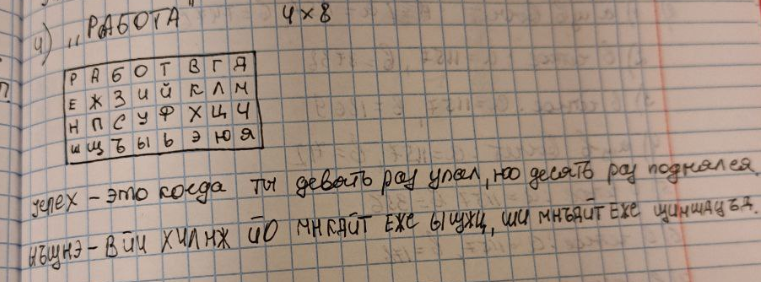
Программная реализация:



Задание 4

Выполните шифрование сообщения «УСПЕХ – ЭТО КОГДА ТЫ ДЕВЯТЬ РАЗ УПАЛ, НО ДЕСЯТЬ РАЗ ПОДНЯЛСЯ», используя систему Трисемуса с ключевым словом из задания 3. Размер таблицы подстановок 4 × 8.

Ручная реализация:



Вывод:

В ходе выполнения лабораторной работы были сформированы умения шифрования с использованием систем Цезаря и системы Трисемуса

Листинг кода

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.Drawing;

using System.Linq;

using System.Security.Policy;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Trisemus

{

public partial class Form1 : Form

{

string[,] Trisemus;

string Alphabet = "абвгдежзийклмнопрстуфхцчшщъыьэюя";

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

int rows = Convert.ToInt32(textBox1.Text);

int cols = Convert.ToInt32(textBox2.Text);

dataGridView1.Rows.Clear();

dataGridView1.ColumnCount = cols;

dataGridView1.RowCount = rows;

Trisemus = new string[rows, cols];

string key\_word = textBox3.Text;

char[] new\_alphabet = Alphabet.ToCharArray();

char[] word\_symbols = key\_word.Distinct().ToArray();

new\_alphabet = new\_alphabet.Except(word\_symbols).ToArray();

new\_alphabet = word\_symbols.Concat(new\_alphabet).ToArray();

int schet = 0;

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

Trisemus[i, j] = new\_alphabet[schet++].ToString();

}

}

for (int i = 0; i < dataGridView1.RowCount; i++)

{

for (int j = 0; j < dataGridView1.ColumnCount; j++)

{

dataGridView1[j, i].Value = Trisemus[i, j];

}

}

string word = textBox4.Text;

string encrypted\_word = "";

foreach (char c in word)

{

if (!Alphabet.Contains(c))

{

encrypted\_word += c;

}

else

{

bool found = false;

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

if (Trisemus[i, j] == c.ToString())

{

found = true;

if (i + 1 == rows)

{

encrypted\_word += Trisemus[0, j];

}

else

{

encrypted\_word += Trisemus[i + 1, j];

}

break;

}

}

if (found)

{

break;

}

}

if (found == false)

{

encrypted\_word += c;

}

}

}

textBox5.Text = encrypted\_word;

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

int rows = Convert.ToInt32(textBox1.Text);

int cols = Convert.ToInt32(textBox2.Text);

dataGridView1.Rows.Clear();

dataGridView1.ColumnCount = cols;

dataGridView1.RowCount = rows;

Trisemus = new string[rows, cols];

string key\_word = textBox3.Text;

char[] new\_alphabet = Alphabet.ToCharArray();

char[] word\_symbols = key\_word.Distinct().ToArray();

new\_alphabet = new\_alphabet.Except(word\_symbols).ToArray();

new\_alphabet = word\_symbols.Concat(new\_alphabet).ToArray();

int schet = 0;

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

Trisemus[i, j] = new\_alphabet[schet++].ToString();

}

}

for (int i = 0; i < dataGridView1.RowCount; i++)

{

for (int j = 0; j < dataGridView1.ColumnCount; j++)

{

dataGridView1[j, i].Value = Trisemus[i, j];

}

}

string word = textBox6.Text;

string encrypted\_word = "";

foreach (char c in word)

{

if (!Alphabet.Contains(c))

{

encrypted\_word += c;

}

else

{

bool found = false;

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

if (Trisemus[i, j] == c.ToString())

{

found = true;

if (i - 1 < 0)

{

encrypted\_word += Trisemus[rows - 1, j];

}

else

{

encrypted\_word += Trisemus[i - 1, j];

}

break;

}

}

if (found)

{

break;

}

}

if (found == false)

{

encrypted\_word += c;

}

}

}

textBox5.Text = encrypted\_word;

}

private void textBox6\_TextChanged(object sender, EventArgs e)

{

}

}

} using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace CesarWithWord

{

public partial class Form1 : Form

{

string Alphabet = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя";

string new\_alpha;

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

string word = textBox1.Text;

word = new string(word.ToLower().Distinct().ToArray());

int k = Convert.ToInt32(textBox2.Text);

char[] new\_alphabet = new char[Alphabet.Length];

if (k < Alphabet.Length - 1)

{

foreach (char c in word)

{

if (!new\_alphabet.Contains(c))

{

new\_alphabet[k] = c;

k = (k + 1) % new\_alphabet.Length;

}

}

foreach (char c in Alphabet)

{

if (!new\_alphabet.Contains(c))

{

new\_alphabet[k] = c;

k = (k + 1) % new\_alphabet.Length;

}

}

new\_alpha = new string(new\_alphabet);

for (int i = 0; i < Alphabet.Length; i++)

{

string letter = Alphabet[i] + " =>" + new\_alpha[i];

listBox1.Items.Add(letter);

}

}

else

{

MessageBox.Show("Ключ k не соблюдает условия");

}

}

private void button2\_Click(object sender, EventArgs e)

{

string new\_word = textBox3.Text;

new\_word = new\_word.ToLower();

string encypted\_word = "";

foreach (char c in new\_word)

{

if (!Alphabet.Contains(c))

{

encypted\_word += c;

}

else

{

int position = Alphabet.IndexOf(c);

char enc\_char = new\_alpha[position];

encypted\_word += enc\_char;

}

}

textBox4.Text = encypted\_word;

}

private void button3\_Click(object sender, EventArgs e)

{

string new\_word = textBox5.Text;

new\_word = new\_word.ToLower();

string decrypted\_word = "";

foreach (char c in new\_word)

{

if (!Alphabet.Contains(c))

{

decrypted\_word += c;

}

else

{

int position = new\_alpha.IndexOf(c);

Debug.WriteLine(position);

char enc\_char = Alphabet[position];

Debug.WriteLine(enc\_char);

decrypted\_word += enc\_char;

}

}

Debug.WriteLine(decrypted\_word);

textBox4.Text = decrypted\_word;

}

}

}