**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное учреждение высшего образования

**«Национальный исследовательский Томский политехнический университет»**

Инженерная школа информационных технологий и робототехники

Отделение информационных технологий

Направление «Информационные системы и технологии»

Отчёт по лабораторной работе №3

по дисциплине «**Информационная безопасность и защита информации**»

**Шифр, основанный на методе умножения матриц**

Выполнил:

Студент группы 8И7Б \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Карташев И.А.

Проверил:

Ассистент ОИТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Григорьев Д. С.

Томск 2021

# Задание

1. Заполнить таблицу 2 в массив, в котором должны храниться все буквы русского алфавита от а до я и от А до Я плюс символы пробел, точка, двоеточие, восклицательный знак, вопросительный знак и запятая (всего 72 символа).
2. Зашифровать любое сообщение, введенное с клавиатуры, методом произведения матриц.
3. Определить какой должна быть матрица, чтобы зашифрованную фразу можно было расшифровать.
4. Расшифровать сообщение.

**Ход работы**

Программа реализована на языке C#, она представляет собой консольное приложение. В программе имеется две опции, расшифровать строку или зашифровать строку.

# Листинг программы

using System;

using System.Collections.Generic;

namespace lab\_3\_\_\_infbez

{

class Program

{

static void Main(string[] args)

{

static int[] Gauss(int[,] key, double[,] key2)

{

int n = key.GetLength(0); //Размерность начальной матрицы (строки)

double[,] key1 = new double[n,n+1];

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

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

key1[i, j] = key[i, j];

//Прямой ход (Зануление нижнего левого угла)

for (int k = 0; k < n; k++) //k-номер строки

{

for (int i = 0; i < n + 1; i++) //i-номер столбца

key2[k, i] = key2[k, i] / key1[k, k]; //Деление k-строки на первый член !=0 для преобразования его в единицу

for (int i = k + 1; i < n; i++) //i-номер следующей строки после k

{

double K = key2[i, k] / key2[k, k]; //Коэффициент

for (int j = 0; j < n + 1; j++) //j-номер столбца следующей строки после k

key2[i, j] = key2[i, j] - key2[k, j] \* K; //Зануление элементов матрицы ниже первого члена, преобразованного в единицу

}

for (int i = 0; i < n; i++) //Обновление, внесение изменений в начальную матрицу

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

key1[i, j] = key2[i, j];

}

//Обратный ход (Зануление верхнего правого угла)

for (int k = n - 1; k > -1; k--) //k-номер строки

{

for (int i = n; i > -1; i--) //i-номер столбца

key2[k, i] = key2[k, i] / key1[k, k];

for (int i = k - 1; i > -1; i--) //i-номер следующей строки после k

{

double K = key2[i, k] / key2[k, k];

for (int j = n; j > -1; j--) //j-номер столбца следующей строки после k

key2[i, j] = key2[i, j] - key2[k, j] \* K;

}

}

//Отделяем от общей матрицы ответы

int[] Answer = new int[n];

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

{

if (i % 3 == 0)

Answer[i] = (int)Math.Round(key2[i, n] + 1);

Answer[i] = (int)Math.Round(key2[i, n]);

}

return Answer;

}

char[,] str = {

{'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж'},

{'З', 'И', 'Й', 'К', 'Л', 'М', 'Н', 'О'},

{'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц'},

{'Ч', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь', 'Э', 'Ю'},

{'Я', 'а', 'б', 'в', 'г', 'д', 'е', 'ё'},

{'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н'},

{'о', 'п', 'р', 'с', 'т', 'у', 'ф', 'х'},

{'ц', 'ч', 'ш', 'щ', 'ъ', 'ы', 'ь', 'э'},

{'ю', 'я', ' ', '.', ':', '!', '?', ','},

};

Console.WriteLine("Зашифровать строку: введите 1");

Console.WriteLine("Расшифровать строку: введите 2");

Console.WriteLine("Закончить: введите 0");

int[,] key = { { 1, 3, 2 }, { 2, 6, 4 }, { 3, 2, 1 } };

string run = "";

run = Console.ReadLine();

while (run != "0")

{

if (run == "1")

{

Console.WriteLine("Введите строку:");

string input = Console.ReadLine();

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

List<int> code2 = new List<int>();

bool sucsessFlag = true;

int k = 0;

for (int m = 0; m < input.Length; m++)

{

bool flag = true;

int i = 0;

int j = 0;

while (flag && (i < 9))

{

while (flag && (j < 8))

{

if (input[m] == str[i, j])

{

k = m / 3;

int h = m % 3;

code.Add((i + 1) \* 10 + j + 1);

flag = false;

}

j++;

}

i++;

j = 0;

}

if (flag)

{

sucsessFlag = false;

Console.WriteLine("В ведённой строке присутствует недопустимый символ, попробуйте снова\n");

break;

}

}

if (code.Count % 3 == 1)

{ code.Add(93); code.Add(93); }

if (code.Count % 3 == 2)

code.Add(93);

k = 0;

for (int m = 0; m < code.Count; m++)

{

int j = 0;

int i = m % 3;

if (m % 3 == 0 && m != 0)

k += 3;

code2.Add(key[i, j] \* code[k] + key[i, j + 1] \* code[k + 1] + key[i, j + 2] \* code[k + 2]);

}

if (sucsessFlag)

{

Console.WriteLine("Зашифрованная строка:");

for (int i = 0; i < code2.Count; i++)

Console.Write(code2[i] + " ");

Console.WriteLine(); Console.WriteLine();

}

}

if (run == "2")

{

Console.WriteLine("Введите зашифрованную строку:");

string[] output = Console.ReadLine().Split(' ');

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

List<int> code2 = new List<int>();

bool sucsessFlag = true;

for (int m = 0; m < output.Length; m++)

{

code.Add(int.Parse(output[m]));

}

if ((key[0,0] \* key[1,1] \* key[2,2] + key[0, 1] \* key[1, 2] \* key[2, 0] + key[0, 2] \* key[1, 0] \* key[2, 1] -

key[0, 2] \* key[1, 1] \* key[2, 0] - key[0, 1] \* key[1, 0] \* key[2, 2] - key[0, 0] \* key[1, 2] \* key[2, 1]) == 0)

sucsessFlag = false;

if (sucsessFlag)

{

for (int m = 0; m < code.Count - 2; m += 3)

{

double[,] key2 = { { 1, 3, 2, code[m] }, { 2, 6, 4, code[m + 1] }, { 3, 2, 1, code[m + 2] } };

int[] answer = Gauss(key, key2);

for (int i = 0; i < 3; i++)

code2.Add(answer[i]);

}

Console.WriteLine("Расшифрованная строка: ");

for (int i = 0; i < code2.Count; i++)

{

int indI = code2[i] / 10 - 1;

int indJ = code2[i] % 10 - 1;

Console.Write(str[indI, indJ]);

}

Console.WriteLine(); Console.WriteLine();

}

else

{

Console.WriteLine("Матрица вырождена (детерминант = 0), задайте другую\n");

break;

}

}

Console.WriteLine("Зашифровать строку: введите 1");

Console.WriteLine("Расшифровать строку: введите 2");

Console.WriteLine("Закончить: введите 0");

run = Console.ReadLine();

}

}

}

}

# Результаты работы программы

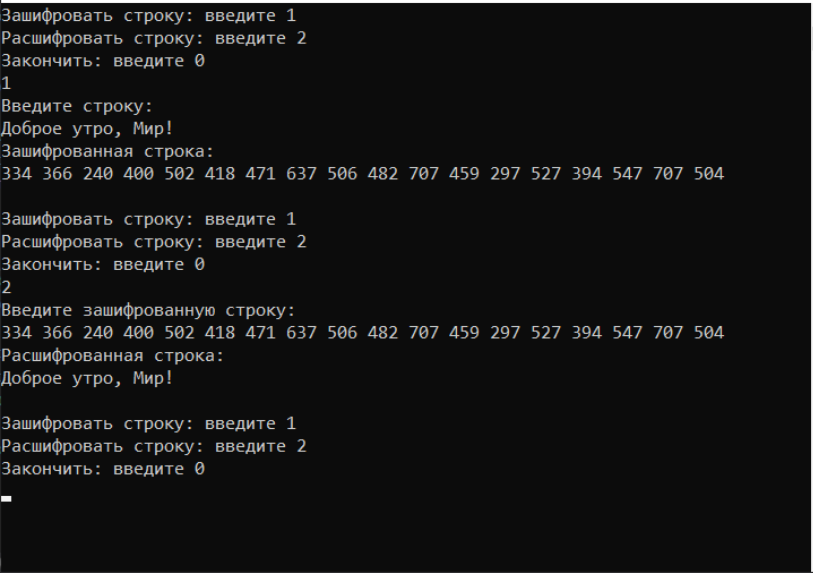


Рисунок 1 – Результат работы программы

# Вывод

В ходе выполнения работы был изучен метод шифрования с помощью перемножения матриц. Была реализована программа, позволяющая зашифровать и расшифровать вводимый пользователем текст, используя данный метод.

Шифр является моноалфавитным, что значит, что злоумышленник может получить доступ к данным, используя метод частотного анализа.

Ссылка на github: https://github.com/Vano-100/inf-security/tree/main/lab3