Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

**Кафедра информационных систем и программной инженерии**

**Лабораторная работа №7**

**по дисциплине**

**«Администрирование и безопасность программно-информационных систем»**

Шифрование с помощью аналитических преобразований

Выполнил:

ПРИ-120

М.А. Бочков

Принял:

доцент каф. ИСПИ

С. В. Курочкин

Владимир 2023

**ЦЕЛЬ РАБОТЫ**

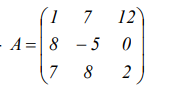
Освоить на практике применение алгоритма шифрования с помощью аналитических преобразований.

**ВЫПОЛНЕНИЕ РАБОТЫ**

**Вариант №6**

**Задание**

Зашифровать описанным методом следующее выражение: «Помехоустойчивое кодирование – это кодирование с возможностью восстановления потерянных или ошибочно принятых данных». В качестве ключа использовать матрицу, представленную на скриншоте №1



Скриншот №1. Исходная матрица

Листинг C#:

using System;

using System.Collections.Generic;

namespace Lab7

{

class Program

{

static void Main(string[] args)

{

string text = "Помехоустойчивое кодирование это кодирование с возможностью восстановления потерянных или ошибочно принятых данных";

Console.Write("Исходный текст: ");

Console.WriteLine(text);

Console.WriteLine("");

text = text.Replace(" ", "").ToLower();

string alph = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя";

int[,] matrix = { { 1, 7, 12 }, { 8, -5, 0 }, { 7, 8, 2 } };

string codeStr = Code(matrix, alph, text);

Console.WriteLine("Закодированный текст:");

Console.Write(codeStr);

}

private static string Code(int[,] matrix, string alph, string text)

{

string codeStr = "";

int temp = 0;

int pointer = 0;

int[,] multMatrix = new int[3, text.Length / 3];

int[,] resultMatrix = new int[3, text.Length / 3];

for (int k = 0; k < text.Length / 3; k++)

{

for (int i = k; i < k + 1; i++)

{

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

{

multMatrix[j, i] = alph.IndexOf(text.Substring(pointer, 1));

pointer += 1;

}

}

}

for (int k = 0; k < text.Length / 3; k++)

{

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

{

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

{

temp += matrix[i, j] \* multMatrix[j, k];

}

resultMatrix[i, k] = temp;

temp = 0;

}

}

for (int k = 0; k < text.Length / 3; k++)

{

for (int i = k; i < k + 1; i++)

{

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

{

codeStr += resultMatrix[j, i].ToString() + ", ";

}

}

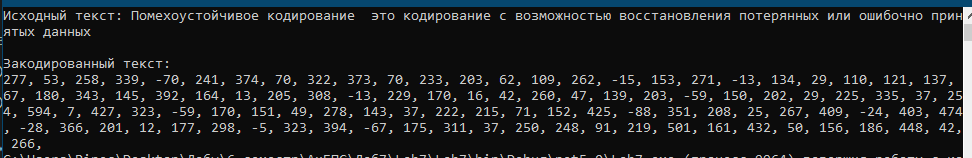
}

return codeStr;

}

}

}



Скриншот №2. Результат работы программы.

**ВЫВОДЫ**

В процессе выполнения работы было выполнено шифрование с помощью аналитических преобразований.