Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Информационные сети. Основы безопасности

ОТЧЕТ

к лабораторной работе №1

на тему

Шифр Цезаря. Шифр Виженера

Выполнил Н. В. Климкович

Проверил Е. А. Лещенко

Минск 2024

СОДЕРЖАНИЕ

[1 Введение 3](#_Toc146752068)

[2 Результат выполнения](#_Toc146752070) 4

[3 Вывод 5](#_Toc146752071)

[Приложение А (обязательное) Листинг кода 6](#_Toc146752073)

[Приложение А (обязательное) Блок-схема алгоритма шифра Цезаря 9](#_Toc146752073)

[Приложение А (обязательное) Блок-схема алгоритма шифра Виженера 10](#_Toc146752073)

**1 ВВЕДЕНИЕ**

Целью данной работы является реализовать программные средства шифрования и дешифрования текстовых файлов при помощи Шифра Цезаря, (шифра сдвига, кода Цезаря) и шифра Виженера.

Шифр Цезаря – это вид шифра подстановки, в котором каждый символ в открытом тексте заменяется символом, находящимся на некотором постоянном числе позиций левее или правее него в алфавите. Например, в шифре со сдвигом 4 А была бы заменена на Г, Б станет Д, и так далее.

Шаг шифрования, выполняемый шифром Цезаря, часто включается как часть более сложных схем, таких как шифр Виженера, и все ещё имеет современное приложение в системе ROT13. Как и все моноалфавитные шифры, шифр Цезаря легко взламывается и не имеет практически никакого применения на практике.

Если сопоставить каждому символу алфавита его порядковый номер (нумеруя с 0), то шифрование и дешифрование можно выразить формулами модульной арифметики:

y=(x+k)\ \mod\ n

x=(y-k+n)\ \mod\ n,

где – символ открытого текста, – символ шифрованного текста, – мощность алфавита, а – ключ.

Шифр Виженера состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая tabula recta или квадрат (таблица) Виженера. Применительно к латинскому алфавиту таблица Виженера составляется из строк по 26 символов, причём каждая следующая строка сдвигается на несколько позиций. Таким образом, в таблице получается 26 различных шифров Цезаря. На каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова.

Расшифровывание производится следующим образом: находим в таблице Виженера строку, соответствующую первому символу ключевого слова; в данной строке находим первый символ зашифрованного текста. Столбец, в котором находится данный символ, соответствует первому символу исходного текста.

**2 РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ**

В ходе выполнения лабораторной работы было разработаны программы, читающие данные из файла и шифрующие (дешифрующие) их с помощью с помощью Шифра Цезаря (Рисунок 1) и шифра Виженера (Рисунок 2).

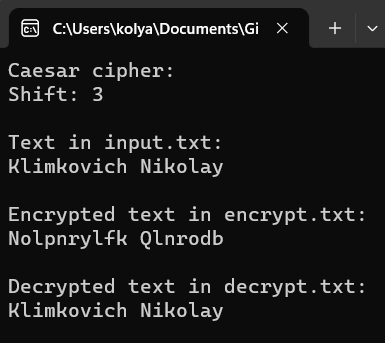


Рисунок 1 – Результат выполнения программы шифра Цезаря

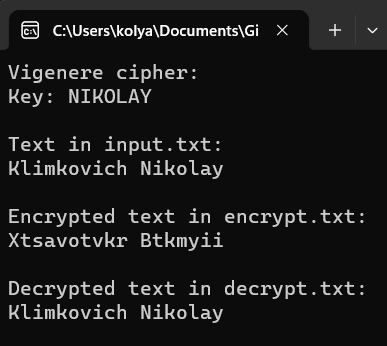


Рисунок 2 – Результат выполнения программы шифра Виженера

В начале каждой программы пользователь вводит необходимые входные данные для дальнейшей работы.

# 3 ВЫВОД

В результате лабораторной работы были разработаны программы, читающие данные из файла и шифрующие (дешифрующие) их с помощью с помощью шифра Цезаря и шифра Виженера.

# ПРИЛОЖЕНИЕ А Листинг кода

Листинг 1 – Файл CaesarCipher.cs

using System.Text;

class CaesarCipher

{

private static string inputFilePath =

Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "input.txt");

private static string encryptFilePath =

Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "encrypt.txt");

private static string decryptFilePath =

Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "decrypt.txt");

static void Main()

{

Console.Write("Caesar cipher:\nShift: ");

uint shift = Convert.ToUInt32(Console.ReadLine()) % 26;

EncryptFile(shift);

DecryptFile(shift);

Console.ReadKey();

}

static void EncryptFile(uint shift)

{

string fileText = File.ReadAllText(inputFilePath);

Console.WriteLine($"\nText in input.txt:\n{fileText}");

string encryptedText = Caesar(fileText, shift);

File.WriteAllText(encryptFilePath, encryptedText);

Console.WriteLine($"\nEncrypted text in encrypt.txt:\n{encryptedText}");

}

static void DecryptFile(uint shift)

{

string encryptedText = File.ReadAllText(encryptFilePath);

string decryptedText = Caesar(encryptedText, 26 - shift);

File.WriteAllText(decryptFilePath, decryptedText);

Console.WriteLine($"\nDecrypted text in decrypt.txt:\n{decryptedText}");

}

static string Caesar(string text, uint shift)

{

StringBuilder result = new StringBuilder();

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

{

if (char.IsLetter(text[i]))

{

char asciiOffset = char.IsUpper(text[i]) ? 'A' : 'a';

result.Append((char)((text[i] + shift - asciiOffset) % 26 + asciiOffset));

}

Else

{

result.Append(text[i]);

}

}

return result.ToString();

}

}

Листинг 2 – Файл VigenereCipher.cs

using System.Text;

class VigenereCipher

{

private static string inputFilePath =

Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "input.txt");

private static string encryptFilePath =

Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "encrypt.txt");

private static string decryptFilePath =

Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "decrypt.txt");

static void Main()

{

Console.Write("Vigenere cipher:\nKey: ");

string key = Console.ReadLine().ToUpper();

EncryptFile(key);

DecryptFile(key);

Console.ReadKey();

}

static void EncryptFile(string key)

{

string fileText = File.ReadAllText(inputFilePath);

Console.WriteLine($"\nText in input.txt:\n{fileText}");

string encryptedText = Vigenere(fileText, key);

File.WriteAllText(encryptFilePath, encryptedText);

Console.WriteLine($"\nEncrypted text in encrypt.txt:\n{encryptedText}");

}

static void DecryptFile(string key)

{

string encryptedText = File.ReadAllText(encryptFilePath);

string decryptedText = Vigenere(encryptedText, key, false);

File.WriteAllText(decryptFilePath, decryptedText);

Console.WriteLine($"\nDecrypted text in decrypt.txt:\n{decryptedText}");

}

static string Vigenere(string text, string key, bool isEncrypt = true)

{

StringBuilder result = new StringBuilder();

int keyIndex = 0;

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

{

if (char.IsLetter(text[i]))

{

char offset = char.IsUpper(text[i]) ? 'A' : 'a';

int keyValue = key[keyIndex] - 'A';

result.Append((char)((text[i] - offset + (isEncrypt ? keyValue : 26 - keyValue)) % 26 + offset));

keyIndex = (keyIndex + 1) % key.Length; }

else {

result.Append(text[i]);

}

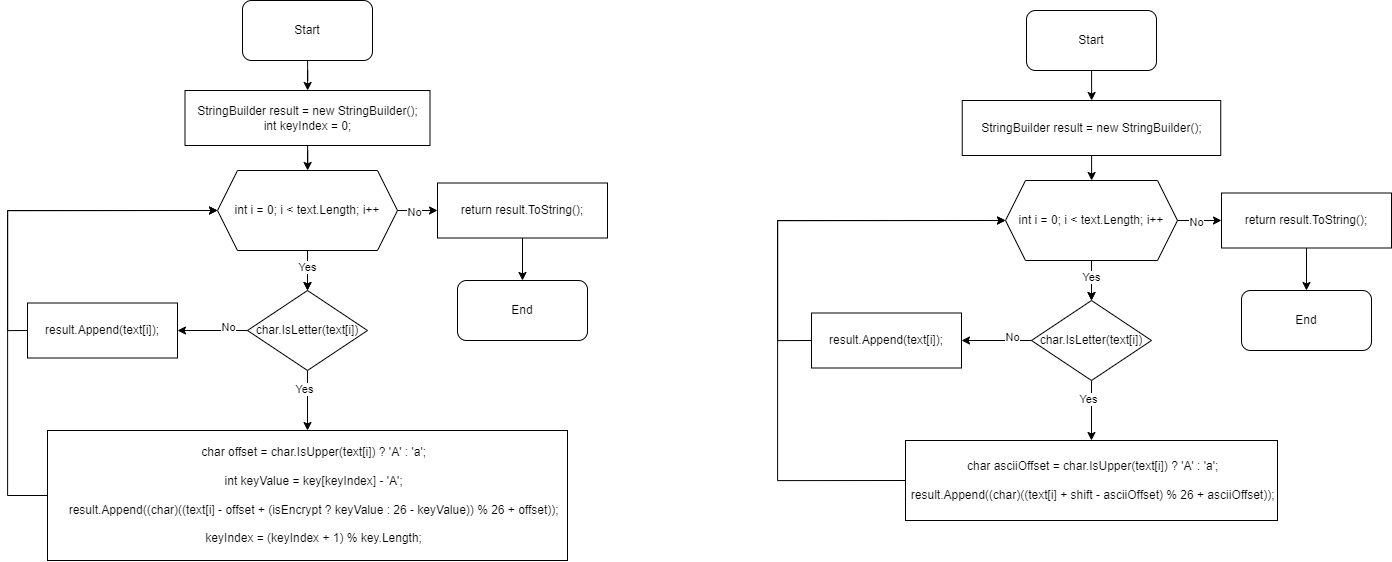
}

return result.ToString();

}

}

# ПРИЛОЖЕНИЕ Б Блок-схема алгоритма шифра Цезаря



# ПРИЛОЖЕНИЕ В Блок-схема алгоритма шифра Виженера

