Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Информационные сети. Основы безопасности

Отчёт

к лабораторной работе

на тему

**ШИФР ЦЕЗАРЯ. ШИФР ВИЖЕНЕРА**

Студент: гр.153501

Савончик Е. В.

Проверил: Лещенко Е. А.

Минск 2024

**СОДЕРЖАНИЕ**

[1 Цель работы 3](#_30j0zll)

[2 Проектирование алгоритма 4](#_1fob9te)

[3 Демонстрация программного средства 5](#_3znysh7)

[Приложение А Блок-схема функции шифрования с помощью шифра Цезаря 6](#_2et92p0)

[Приложение Б Блок-схема функции дешифрования с помощью шифра  
 Цезаря 7](#_tyjcwt)

[Приложение В Блок-схема функции шифрования с помощью шифра  
 Виженера 8](#_3dy6vkm)

[Приложение Г Блок-схема функции дешифрования с помощью шифра  
 Виженера 9](#_1t3h5sf)

[Приложение Д Исходный код программы 10](#_4d34og8)

# 1 ЦЕЛЬ РАБОТЫ

Изучить теоретические сведения о шифре Цезаря и шифре Виженера. Реализовать программные средства шифрования и дешифрования текстовых файлов при помощи шифра Цезаря и шифра Виженера.

# 2 ПРОЕКТИРОВАНИЕ АЛГОРИТМА

Функции шифрования и дешифрования с помощью шифра Цезаря и шифра Виженера весьма похожи и отличаются лишь блоком выбора закодированного символа. Блок-схемы алгоритмов представлены в приложениях A-Г.

# 3 ДЕМОНСТРАЦИЯ ПРОГРАММНОГО СРЕДСТВА

В результате выполнения лабораторной работы было получено программное средство, способное считывать данные из файла и шифровать/дешифровать их. Весь код программы был написан на языке *C#.*

При запуске программы происходит считывание файла и вызов соответствующих функций классов *CaesarCipher* и *VigenereCypher.* Вывод программы после выполнения представлен на рисунках 1, 2, 3, 4.

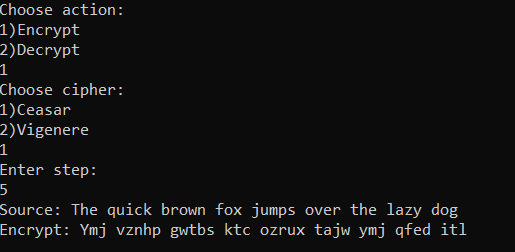
**

Рисунок 1 – Шифрование с помощью шифра Цезаря

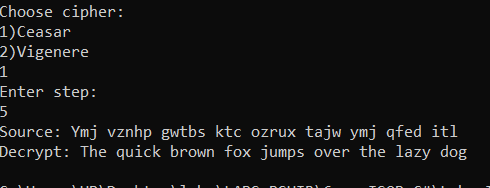


Рисунок 2 – Дешифрование с помощью Шифра Цезаря

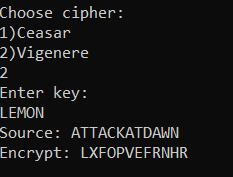


Рисунок 3 – Шифрование с помощью шифра Виженера

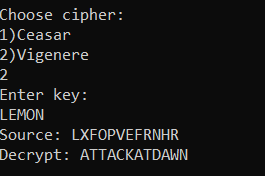
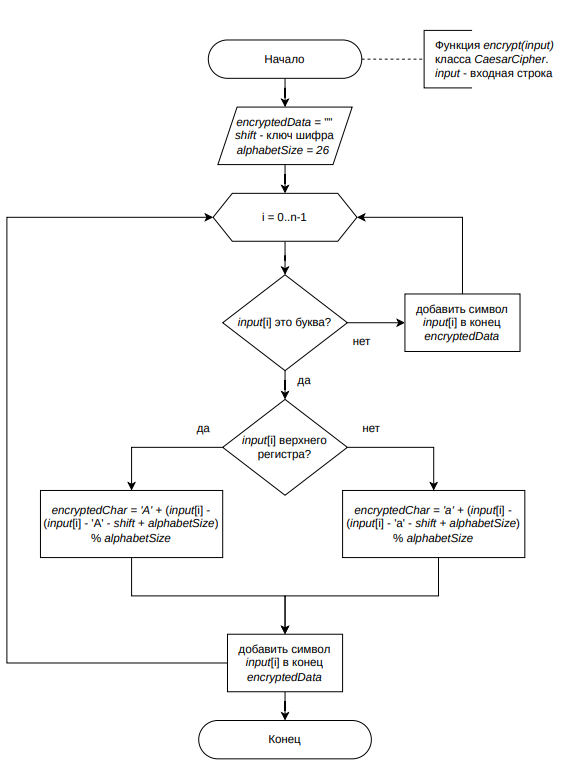


Рисунок 4 – Дешифрование с помощью шифра Виженера

Исходный код программы представлен в приложении Д.

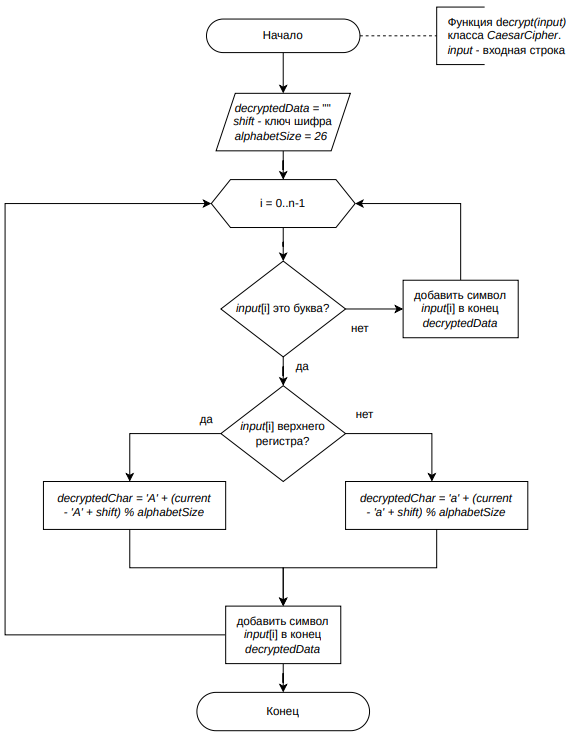
# ПРИЛОЖЕНИЕ А

**Блок-схема функции шифрования с помощью шифра Цезаря**



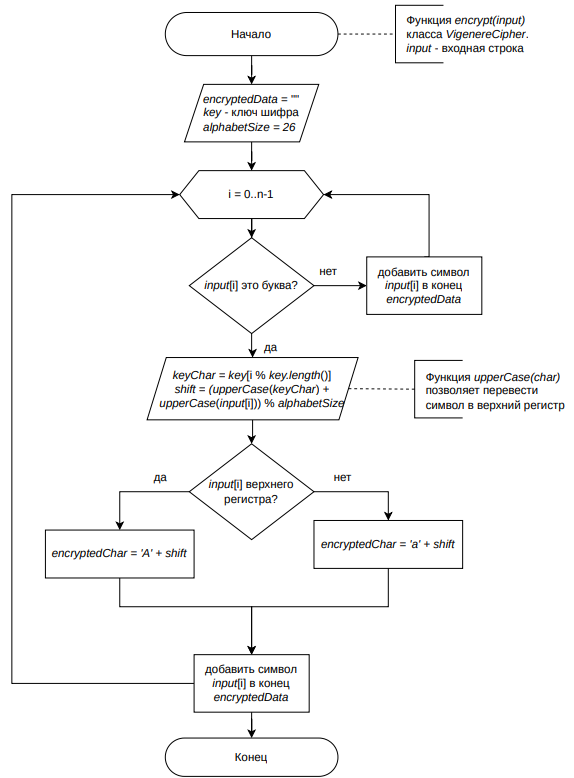
# ПРИЛОЖЕНИЕ Б

**Блок-схема функции дешифрования с помощью шифра Цезаря**



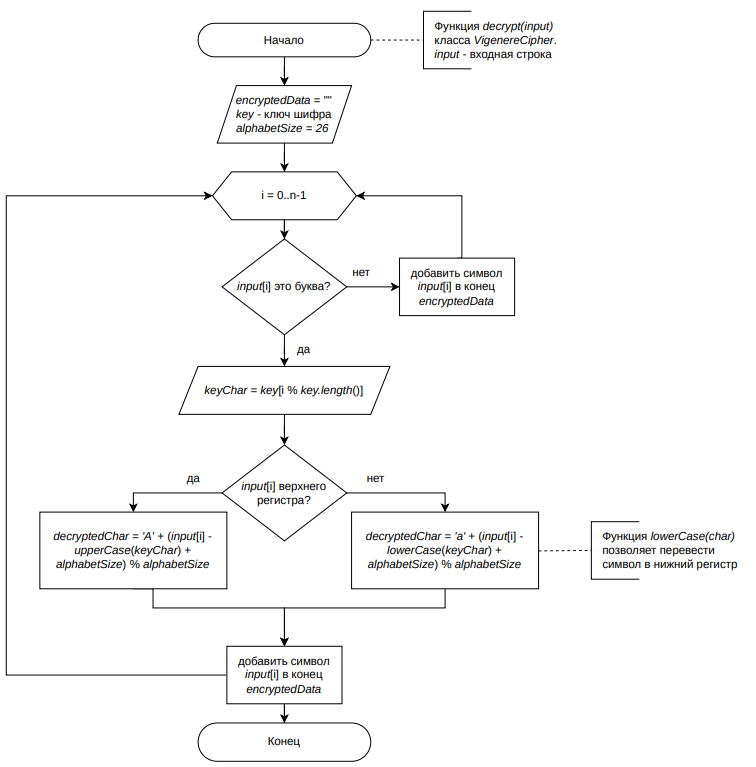
# ПРИЛОЖЕНИЕ В

**Блок-схема функции шифрования с помощью шифра Виженера**



# ПРИЛОЖЕНИЕ Г

**Блок-схема функции дешифрования с помощью шифра Виженера**



# ПРИЛОЖЕНИЕ Д

**Исходный код программы**

Листинг 1 – Файл Programm.cs

namespace Lab\_1

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Test filename: C:\\Users\\HP\\Desktop\\labs\\LABS\_BSUIR\\6sem\_ISOB\_C#\\Labs\_ISOB\\Lab\_1\\Test\\CDecryptTest.txt");

Console.WriteLine("Enter filename: ");

string? fileName = Console.ReadLine();

string source = "";

if (File.Exists(fileName))

{

try

{

source = File.ReadAllText(fileName);

}

catch (IOException e)

{

Console.WriteLine($"Reading file error: {e.Message}");

}

}

else

{

Console.WriteLine("File not exists");

return;

}

Console.WriteLine("Choose action:\n1)Encrypt\n2)Decrypt");

string choiceAction = Console.ReadLine();

Console.WriteLine("Choose cipher:\n1)Ceasar\n2)Vigenere");

string choiceCipher = Console.ReadLine();

if (choiceCipher == "1")

{

Console.WriteLine("Enter step: ");

CaesarСipher cc = new CaesarСipher(Convert.ToInt32(Console.ReadLine()));

Console.WriteLine("Source: " + source);

if(choiceAction == "1")

{

Console.WriteLine("Encrypt: " + cc.Encrypt(source));

}

else if(choiceAction == "2")

{

Console.WriteLine("Decrypt: " + cc.Decrypt(source));

}

}

else if (choiceCipher == "2")

{

Console.WriteLine("Enter key: ");

VigenereСipher vc = new VigenereСipher(Console.ReadLine().Trim());

Console.WriteLine("Source: " + source);

if (choiceAction == "1")

{

Console.WriteLine("Encrypt: " + vc.Encrypt(source));

}

else if (choiceAction == "2")

{

Console.WriteLine("Decrypt: " + vc.Decrypt(source));

}

}

else

{

Console.WriteLine("Invalid value");

}

}

}

}

Листинг 2 – Файл CaesarСipher.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_1

{

public class CaesarСipher

{

public int Step { get; set; }

private const int ALPHABET\_LENGTH = 26;

public CaesarСipher(int step)

{

Step = step;

}

public string Encrypt(string source)

{

StringBuilder result = new StringBuilder();

foreach (char letter in source)

{

if(char.IsLetter(letter) && char.IsUpper(letter))

{

result.Append((char)((letter + Step - 'A') % ALPHABET\_LENGTH + 'A'));

}

else if(char.IsLetter(letter))

{

result.Append((char)((letter + Step - 'a') % ALPHABET\_LENGTH + 'a'));

}

else

{

result.Append(letter);

}

}

return result.ToString();

}

public string Decrypt(string source)

{

StringBuilder result = new StringBuilder();

foreach (char letter in source)

{

if (char.IsLetter(letter) && char.IsUpper(letter))

{

result.Append((char)((letter - Step + ALPHABET\_LENGTH - 'A') % ALPHABET\_LENGTH + 'A'));

}

else if (char.IsLetter(letter))

{

result.Append((char)((letter - Step + ALPHABET\_LENGTH - 'a') % ALPHABET\_LENGTH + 'a'));

}

else

{

result.Append(letter);

}

}

return result.ToString();

}

}

}

Листинг 3 – Файл VigenereСipher.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_1

{

public class VigenereСipher

{

private string \_key;

public string Key

{

get => \_key;

set

{

if (value != null)

\_key = value.ToLower();

}

}

private const int ALPHABET\_LENGTH = 26;

public VigenereСipher(string key)

{

Key = key;

}

public string Encrypt(string source)

{

StringBuilder result = new StringBuilder();

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

{

int letOffset = char.IsUpper(source[i]) ? 'A' : 'a';

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

{

result.Append((char)(((source[i] + Key[i % Key.Length] - letOffset - 'a') % ALPHABET\_LENGTH) + letOffset));

}

}

return result.ToString();

}

public string Decrypt(string source)

{

StringBuilder result = new StringBuilder();

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

{

int letOffset = char.IsUpper(source[i]) ? 'A' : 'a';

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

{

result.Append((char)(((source[i] - Key[i % Key.Length] + ALPHABET\_LENGTH - letOffset + 'a') % ALPHABET\_LENGTH) + letOffset));

}

}

return result.ToString();

}

}

}