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:

студент гр. 253504

Носкович П.Н.

Проверил:

Герчик А.В.

Минск 2025

СОДЕРЖАНИЕ

[Введенеие 3](#_Toc157436533)

[1 Краткие теоретические сведения 4](#_Toc157436534)

[2 Результат выполнения программы 7](#_Toc157436535)

[Приложение А 10](#_Toc157436537)

# ВВЕДЕНИЕ

Шифр – система обратимых преобразований, зависящая от некоторого секретного параметра (ключа) и предназначенная для обеспечения секретности передаваемой информации.

Цель данной лабораторной работы заключается в изучении теоретических сведений по работе Шифра Цезаря и Шифра Виженера и разработки программы, шифрующей и дешифрующей информацию с помощью данных шифров.

# 1 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Шифр Цезаря, также известный, как шифр сдвига, код Цезаря или сдвиг Цезаря – один из самых простых и наиболее широко известных методов шифрования.

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.

Шифр назван в честь римского императора Гая Юлия Цезаря, использовавшего его для секретной переписки со своими генералами.

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.



Рисунок 1 – Шифр Цезаря

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 ~x – символ открытого текста, ~y – символ шифрованного текста, ~n – мощность алфавита, а ~k – ключ.

С точки зрения математики шифр Цезаря является частным случаем аффинного шифра.

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. Например, предположим, что исходный текст имеет такой вид: *ATTACKATDAWN*

Человек, посылающий сообщение, записывает ключевое слово («*LEMON*») циклически до тех пор, пока его длина не будет соответствовать длине исходного текста: *LEMONLEMONLE*

Первый символ исходного текста A зашифрован последовательностью *L*, которая является первым символом ключа. Первый символ *L* шифрованного текста находится на пересечении строки *L* и столбца *A* в таблице Виженера. Точно так же для второго символа исходного текста используется второй символ ключа; то есть второй символ шифрованного текста *X* получается на пересечении строки *E* и столбца *T*. Остальная часть исходного текста шифруется подобным способом.

Исходный текст: *ATTACKATDAWN*; Ключ: *LEMONLEMONLE*; Зашифрованный текст: *LXFOPVEFRNHR*.

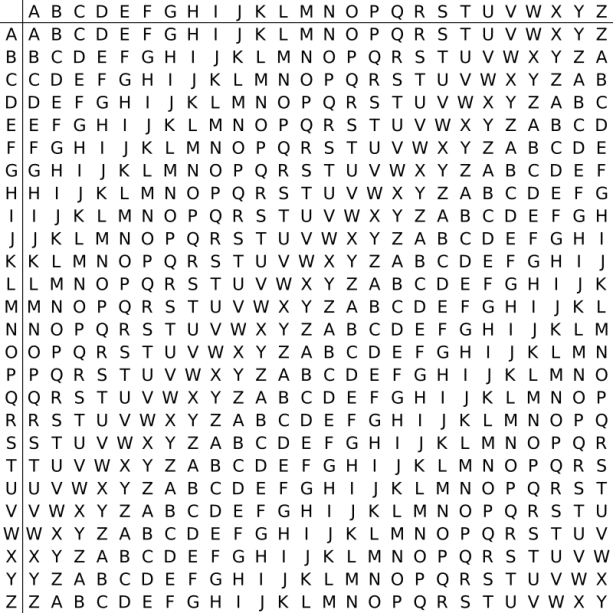


Рисунок 2 – Квадрат Виженера

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. Следующие символы зашифрованного текста расшифровываются подобным образом.

Если *n* — количество букв в алфавите, *m j* — буквы открытого текста, *k j* — буквы ключа, то шифрование Виженера можно записать следующим образом:*c j = m j + k j ( mod n )*.

И расшифровывание: *m j = c j – k j ( mod n ).*

В компьютере такая операция соответствует сложению кодов *ASCII* символов сообщения и ключа по некоторому модулю. Кажется, что если таблица будет более сложной, чем циклическое смещение строк, то шифр станет надежнее. Это действительно так, если ее менять чаще, например, от слова, к слову. Но составление таких таблиц, представляющих собой латинские квадраты.

# 2 РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ

В результате разработки программы было создано консольное приложение, осуществляющее шифровку и дешифровку информации из файлов по заданным ключам.

На рисунке 2.1 представлена блок-схема алгоритма для работы с шифром Цезаря.

На рисунке 2.2 представлена блок-схема алгоритма для работы с шифром Виженера.

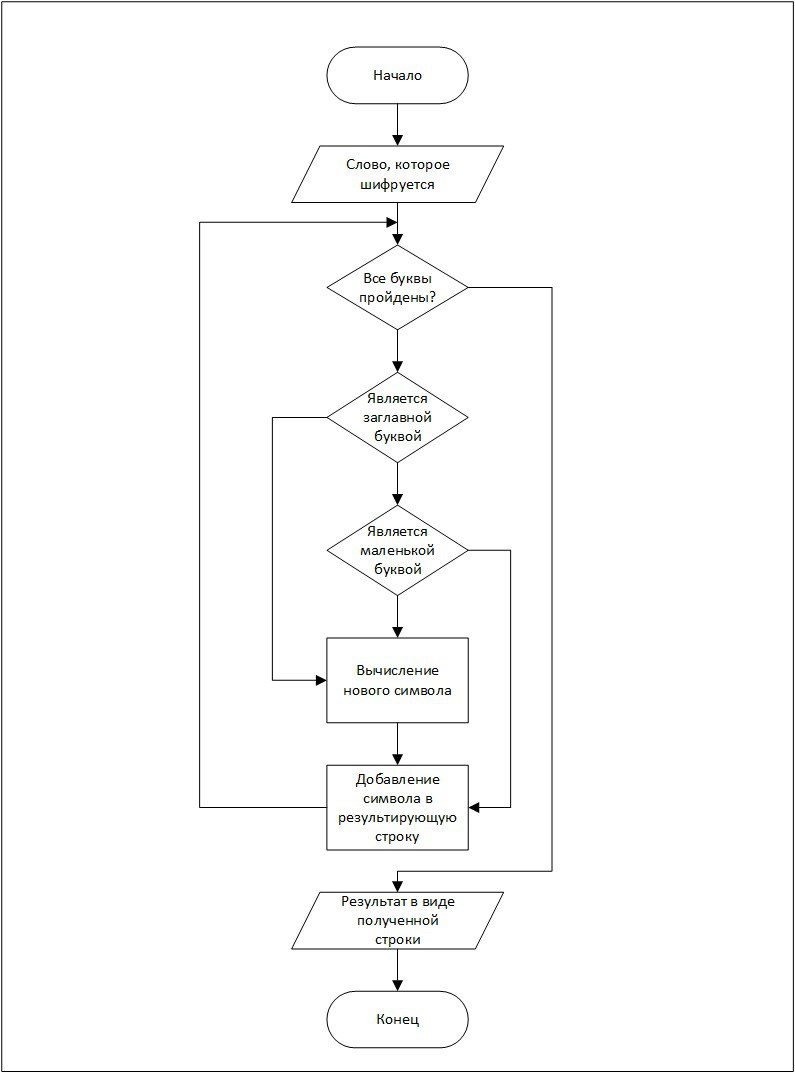


Рисунок 2.1 – Блок-схема алгоритма для работы с шифром Цезаря

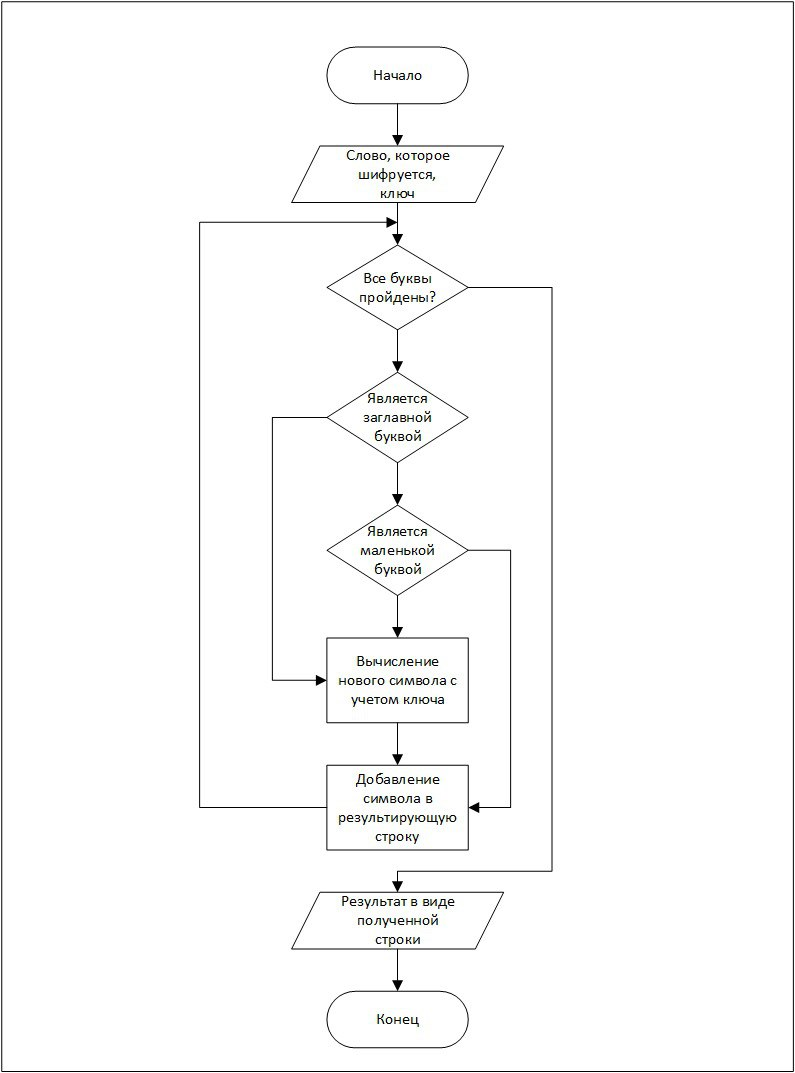


Рисунок 2.2 – Блок-схема алгоритма для работы с шифром Виженера

Результат выполнения программы изображён на рисунке 2.3.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 2.3 – Результат программы

# ПРИЛОЖЕНИЕ А

**(обязательное)**

**Исходный код программы**

**main.py**

def caesar\_cipher(text, shift, encrypt=True):

    result = ''

    for char in text:

        if char.isalpha():

            if 'а' <= char <= 'я' or 'А' <= char <= 'Я':

                start = ord('а') if char.islower() else ord('А')

                alphabet\_size = 32

            else:

                start = ord('a') if char.islower() else ord('A')

                alphabet\_size = 26

            shifted\_char = chr((ord(char) - start + (shift if encrypt else -shift)) % alphabet\_size + start)

        elif char.isdigit():

            shifted\_char = str((int(char) + (shift if encrypt else -shift)) % 10)

        else:

            shifted\_char = char  # Не изменяем пробелы и знаки препинания

        result += shifted\_char

    return result

def vigenere\_cipher(text, key, encrypt=True):

    result = ''

    key\_len = len(key)

    key\_index = 0

    for char in text:

        if char.isalpha():

            if 'а' <= char <= 'я' or 'А' <= char <= 'Я':

                start = ord('а') if char.islower() else ord('А')

                alphabet\_size = 32

            else:

                start = ord('a') if char.islower() else ord('A')

                alphabet\_size = 26

            key\_char = key[key\_index % key\_len].lower()

            if 'а' <= key\_char <= 'я':

                shift = ord(key\_char) - ord('а')

            else:

                shift = ord(key\_char) - ord('a')

            # Для расшифровки мы используем отрицательный сдвиг

            shifted\_char = chr((ord(char) - start - (shift if encrypt else -shift)) % alphabet\_size + start)

            key\_index += 1

        else:

            shifted\_char = char  # Не изменяем пробелы и знаки препинания

        result += shifted\_char

    return result

def get\_valid\_choice(prompt, valid\_choices):

    while True:

        choice = input(prompt)

        if choice in valid\_choices:

            return choice

        else:

            print(f"Ошибка! Введите одно из значений: {', '.join(valid\_choices)}")

def get\_valid\_int\_input(prompt):

    while True:

        try:

            value = int(input(prompt))

            return value

        except ValueError:

            print("Ошибка! Пожалуйста, введите целое число.")

def get\_valid\_string\_input(prompt):

    while True:

        value = input(prompt).strip()

        if value:  # Убедимся, что строка не пустая

            return value

        else:

            print("Ошибка! Пожалуйста, введите не пустую строку.")

def main():

    input\_text = get\_valid\_string\_input("Введите текст для обработки: ")

    action\_choice = get\_valid\_choice("Выберите действие (1 - Зашифровать, 2 - Расшифровать): ", ["1", "2"])

    if action\_choice == "1":

        cipher\_choice = get\_valid\_choice("Выберите метод шифрования (1 - Цезарь, 2 - Виженер): ", ["1", "2"])

        if cipher\_choice == "1":

            key = get\_valid\_int\_input("Введите ключ для шифра Цезаря (целое число): ")

            ciphered\_text = caesar\_cipher(input\_text, key)

            print("\nЗашифрованный текст:   " + ciphered\_text + '\n')

        elif cipher\_choice == "2":

            key = get\_valid\_string\_input("Введите ключ для шифра Виженера (строка): ")

            ciphered\_text = vigenere\_cipher(input\_text, key)

            print("\nЗашифрованный текст:   " + ciphered\_text + '\n')

    elif action\_choice == "2":

        cipher\_choice = get\_valid\_choice("Выберите метод расшифрования (1 - Цезарь, 2 - Виженер): ", ["1", "2"])

        if cipher\_choice == "1":

            key = get\_valid\_int\_input("Введите ключ для шифра Цезаря (целое число): ")

            deciphered\_text = caesar\_cipher(input\_text, key, encrypt=False)

            print("\nРасшифрованный текст:   " + deciphered\_text + '\n')

        elif cipher\_choice == "2":

            key = get\_valid\_string\_input("Введите ключ для шифра Виженера (строка): ")

            deciphered\_text = vigenere\_cipher(input\_text, key, encrypt=False)

            print("\nРасшифрованный текст:   " + deciphered\_text + '\n')

if \_\_name\_\_ == "\_\_main\_\_":

    main()