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**

Студент Е. В. Ласевич

Преподаватель Е. А. Лещенко

Минск 2024

**СОДЕРЖАНИЕ**

[1 Постановка задачи 3](#_6127s8b6t8wy)

[2 Блок-схема](#_r4y5108cx8yu) алгоритма [4](#_r4y5108cx8yu)

[3 Результат выполнения лабораторной работы](#_jnfz1qbvv5j4) 6

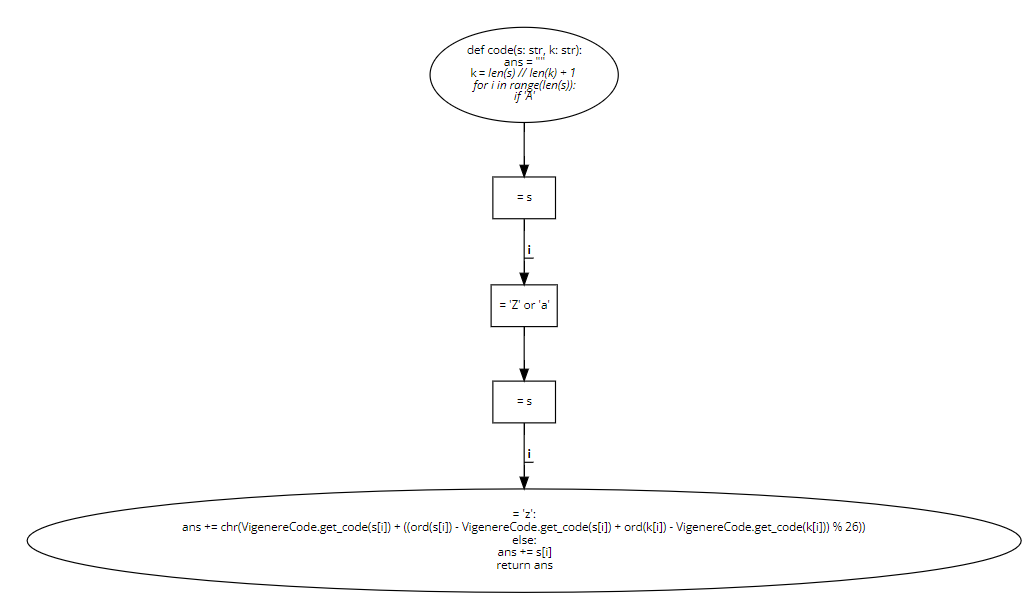
[Выводы](#_ldsbh3e5c0xh) 7

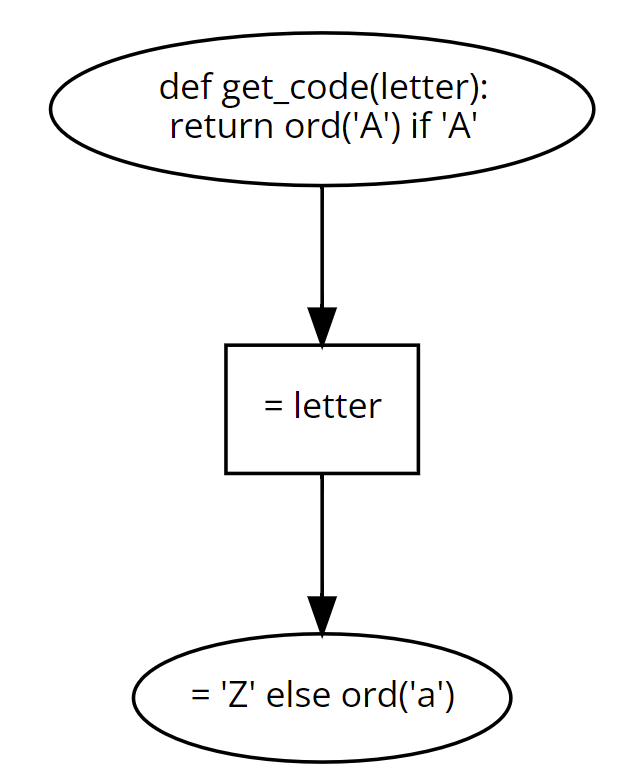
[Приложение А (обязательное) Листинг кода](#_54xlhz7mfhe) 8

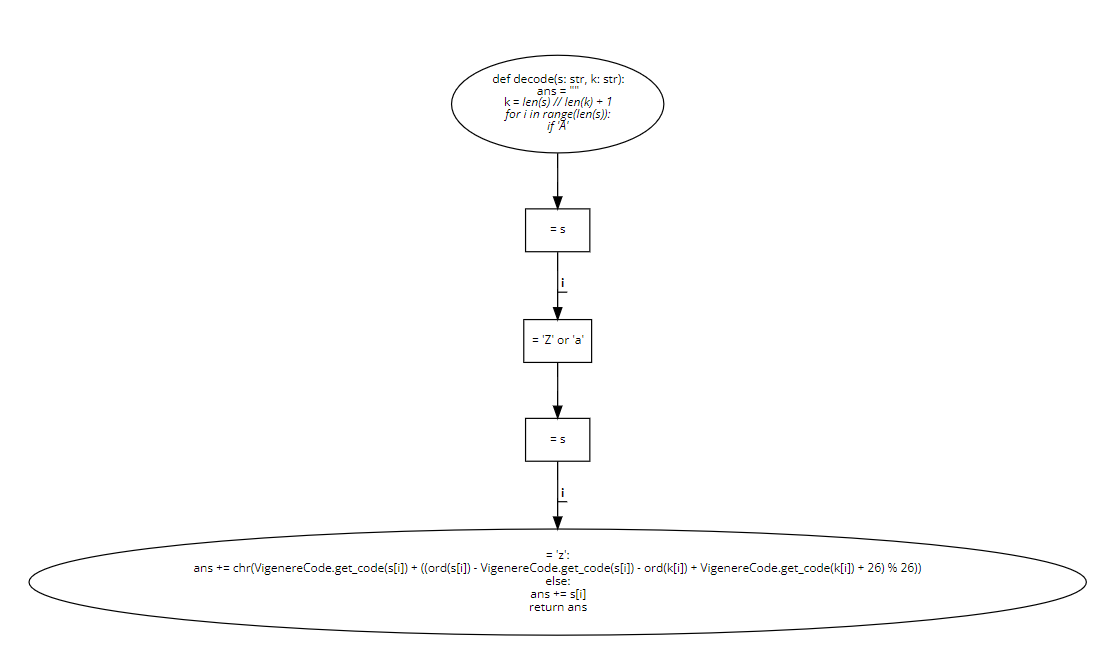
# 1 ПОСТАНОВКА ЗАДАЧИ

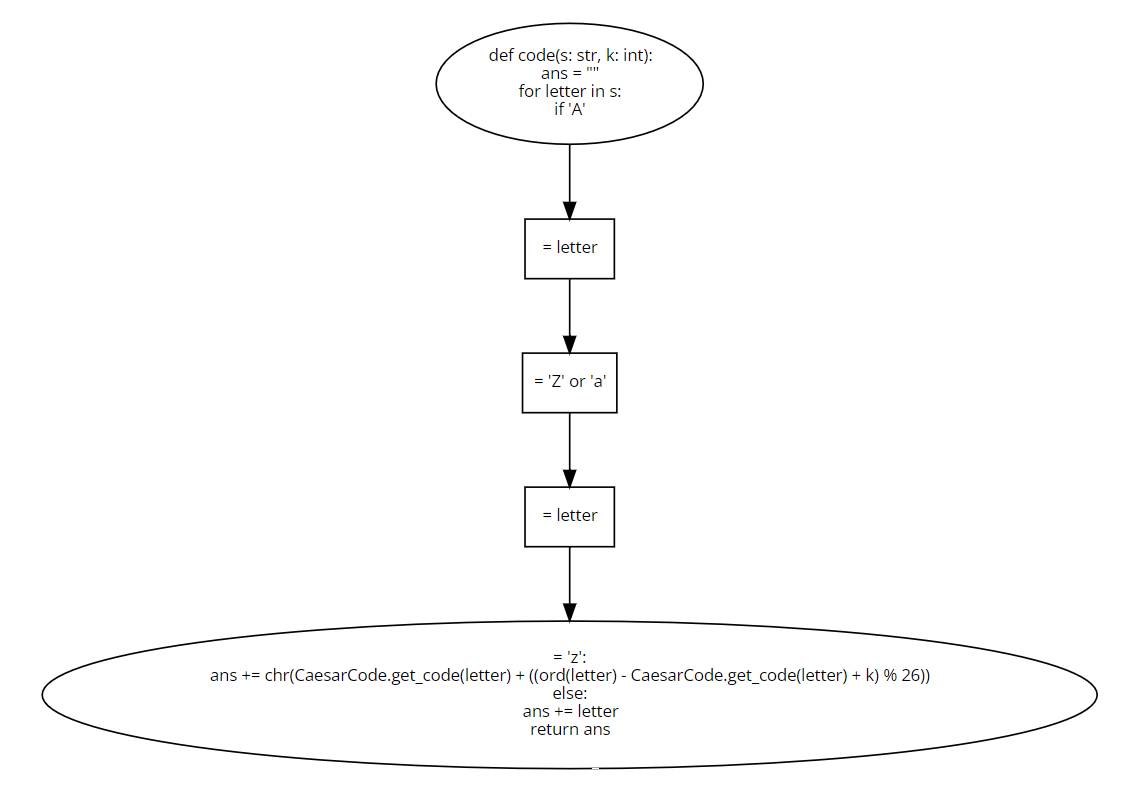
Целью выполнения данной лабораторной работы является разработка программных средств для шифрования и дешифрования текстовых файлов, используя метод Шифра Цезаря (также известного как шифр сдвига или код Цезаря) и метод шифрования Виженера.

# 2 БЛОК-СХЕМА АЛГОРИТМА9









# 

# 3 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

Изначально имеется файл, содержащий строку: “ The quick brown fox jumps over the lazy dog. 123”.

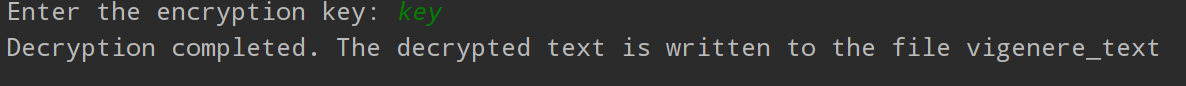


Рисунок 1 ­­­­– Шифрование Виженера

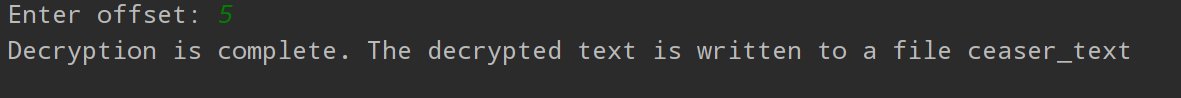


Рисунок 2 ­­­­– Шифрование Цезаря

# ВЫВОДЫ

Для лабораторной работы по теме “Шифр Цезаря” и “Шифр Виженера” был написан код на языке Python. Для шифра Цезаря были реализованы функции code и decode, которые принимают строку и число, и возвращают новую строку, где каждая буква исходной строки сдвигается на k позиций в алфавите. Если символ не является буквой, он остается без изменений. Для шифра Виженера была реализована функция code, которая принимает строку и ключ (также строку), и возвращает новую строку, где каждая буква исходной строки сдвигается на количество позиций в алфавите, соответствующее букве ключа. Если символ не является буквой, он остается без изменений.

# ПРИЛОЖЕНИЕ А

# (обязательное)

# Листинг кода

Листинг 1 – Файл vigenere.py

def encrypt\_vigenere(input\_file, output\_file, key, decrypt=False):  
 with open(input\_file, 'r') as file:  
 plain\_text = file.read()  
  
 encrypted\_text = ""  
 key\_length = len(key)  
 key\_index = 0  
  
 for char in plain\_text:  
 if char.isalpha():  
 key\_char = key[key\_index % key\_length].lower()  
 shift = ord(key\_char) - ord('a')  
 key\_index += 1  
  
 # Determine the direction of the shift based on whether it's encryption or decryption  
 direction = 1 if not decrypt else -1  
  
 if char.isupper():  
 encrypted\_char = chr((ord(char) - ord('A') + direction \* shift) % 26 + ord('A'))  
 else:  
 encrypted\_char = chr((ord(char) - ord('a') + direction \* shift) % 26 + ord('a'))  
 encrypted\_text += encrypted\_char  
 else:  
 encrypted\_text += char  
  
 with open(output\_file, 'w') as file:  
 file.write(encrypted\_text)

Листинг 2 – Файл ceaser.py

def encrypt\_caesar(input\_file, output\_file, shift, decrypt=False):  
 with open(input\_file, 'r') as file:  
 plain\_text = file.read()  
  
 encrypted\_text = ""  
  
 # Determine the direction of the shift based on whether it's encryption or decryption  
 direction = 1 if not decrypt else -1  
  
 for char in plain\_text:  
 if char.isalpha():  
 if char.isupper():  
 encrypted\_char = chr((ord(char) - ord('A') + direction \* shift) % 26 + ord('A'))  
 else:  
 encrypted\_char = chr((ord(char) - ord('a') + direction \* shift) % 26 + ord('a'))  
 encrypted\_text += encrypted\_char  
 else:  
 encrypted\_text += char  
  
 with open(output\_file, 'w') as file:  
 file.write(encrypted\_text)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 input\_file = "text"  
 output\_file = "ceaser\_text"  
 new\_file = "ceaser\_new\_file"  
 shift = int(input("Enter offset: "))  
  
 encrypt\_caesar(input\_file, output\_file, shift, decrypt=False)  
 encrypt\_caesar(output\_file, new\_file, shift, decrypt=True)  
 print("Decryption is complete. The decrypted text is written to a file", output\_file)