Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования   
«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ

Школа профессионального и академического образования

Оценка

**ЛАБОРАТОРНАЯ РАБОТА №1**

**ПО ДИСЦИПЛИНЕ**

«Криптографические методы защиты информации»

Преподаватель Агафонов А.В.

(дата) (подпись) Соколов И.П.

Студент Черепанов А.И.

Группа: РИ – 481223

Екатеринбург

2021

**ЦЕЛЬ РАБОТЫ**

Изучить исторически значимые методы шифрования на основе замен и их недостатки.

**ПРАКТИЧЕСКАЯ ЧАСТЬ**

1. Создать программу, которая зашифровывает и расшифровывает файлы с использованием шифра Цезаря (с. 11 пособия «Основы криптографии» Алферова и др., далее - пособия). При этом имена входного и выходного файлов, а также режим работы (зашифрование/расшифрование) и ключ (в качестве которого используется величина смещения алфавита) задаются параметрами командной строки программы (подсказка: sys.argv).

Создадим класс CesarCipher. При инициализации объекта класса создаем 4 списка с алфавитами русским и английским в двух регистрах. А также инициализируем поля **inpfile** – значением имени входного файла для чтения данных, **outfile** – значением имени выходного файла для записи данных, поле **shift** – значением ключа (числовой сдвиг шифра):

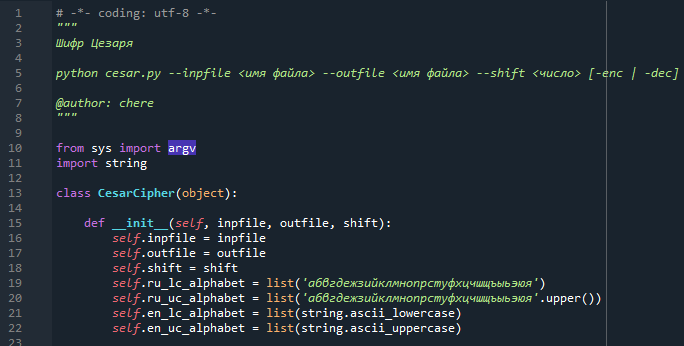


Рисунок 1 – Инициализация CesarCipher

Создадим методы шифрования и дешифрования **encrypt** и **decrypt**. Методы вызываются без аргументов.

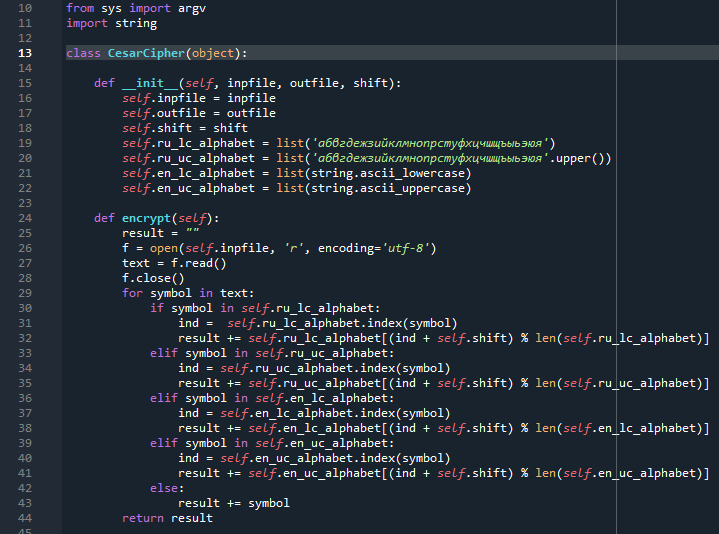


Рисунок 2 – Метод encrypt

При выполнении шифрования в методе encrypt мы прибавляем сдвиг к каждому символу текста и сохраняем получившийся символ в результирующую переменную.

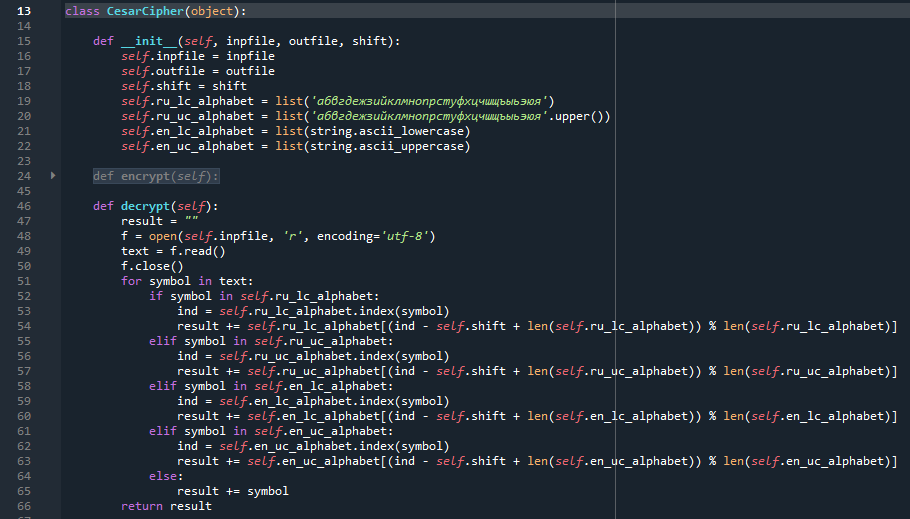


Рисунок 3 – Метод decrypt

При выполнении дешифрования в методе decrypt мы вычитаем сдвиг из каждого символа текста и сохраняем получившийся символ в результирующую переменную.

Перебор аргументов в main() реализован в цикле. Если последний аргумент равен **–enc**, то будет вызван метод **encrypt**, если **–dec**, то **decrypt**.Результат работы методов будет записан в выходной файл.

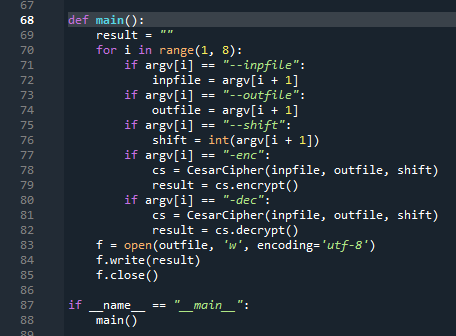
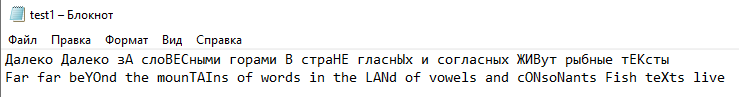


Рисунок 4 – Процедура main()

2. Зашифровать и расшифровать произвольный текстовый файл.

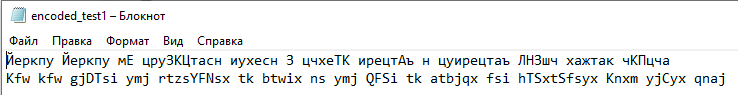
Запишем во входном файле **test1.txt** две строки текста, одна из которой содержит буквы русского алфавита, а вторая – английского:



Затем вызовем из командной строки программу с соответствующими аргументами:

python cesar.py --inpfile test1.txt --outfile encoded\_test1.txt --shift 5 -enc

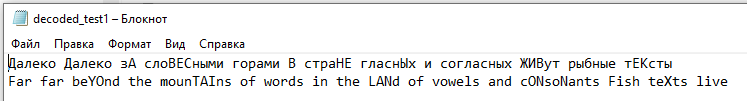
Получим в файле **encoded\_test1.txt** зашифрованный текст:



Затем снова вызовем программу, но в качестве входного файла укажем **encoded\_test1.txt**

python cesar.py --inpfile encoded\_test1.txt --outfile decoded\_test1.txt --shift 5 -dec

Расшифрованный текст будет записан в файл **decoded\_test1.txt**

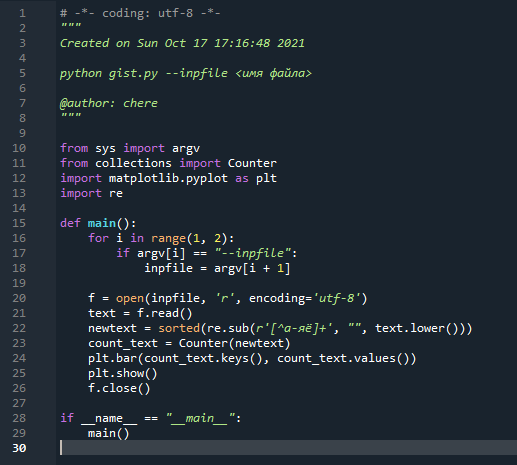


Можно убедиться, что исходный и расшифрованный текст совпадают, следовательно, оба метода класса CesarCipher работают верно.

3. Изучить главу 7 пособия.

Глава 7 была изучена.

4. Выполнить криптоанализ на основе частот появления символов: построить 2 гистограммы, показывающие доли каждого из символов: 1) в каком-либо объемном текстовом файле (предлагается использовать текст «Войны и мира» Л.Н. Толстого из прилагаемого файла) и 2) в зашифрованном файле. Убедиться в совпадении кодов наиболее часто встречающихся зашифрованных символов с соответствующими кодами открытого текста.



Для примера приведем весь текст к нижнему регистру. Функция **Counter** из модуля **collections**, которая подсчитает частоты появления каждого символа в массиве.

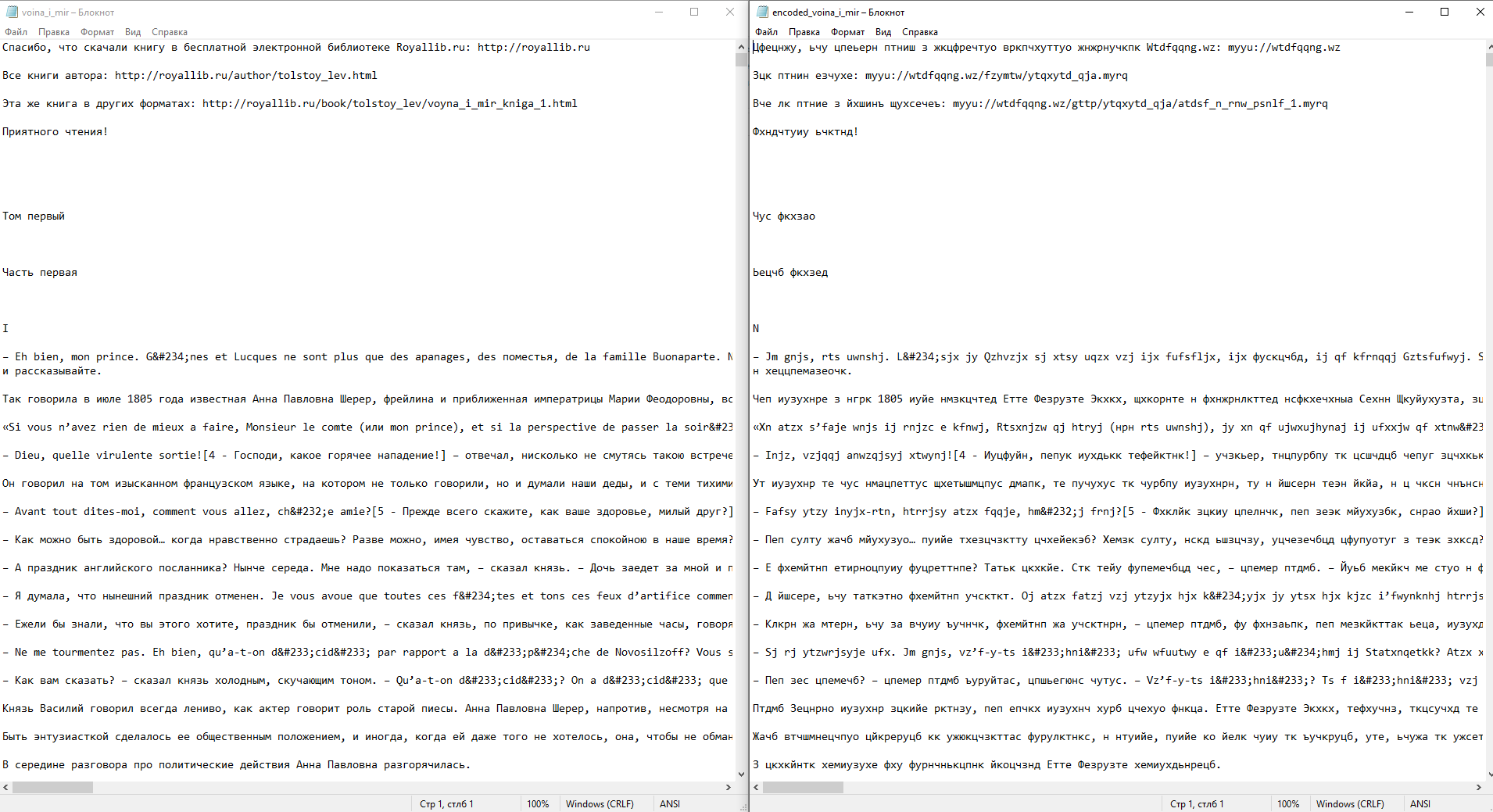


Рисунок 5 – содержимое исходного и зашифрованного файла

В результате работы программы получим две гистограммы, показывающие доли символов исходного и зашифрованного файла соответственно. При запуске был использован ключ, равный 5.

Построим столбчатую диаграмму с помощью метода **bar** из **matplotlib.pyplot**:

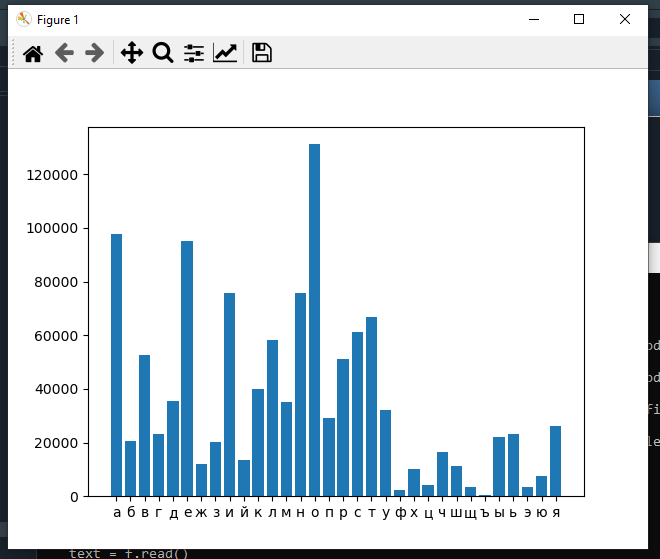


Рисунок 6 – доля символов в исходном файле

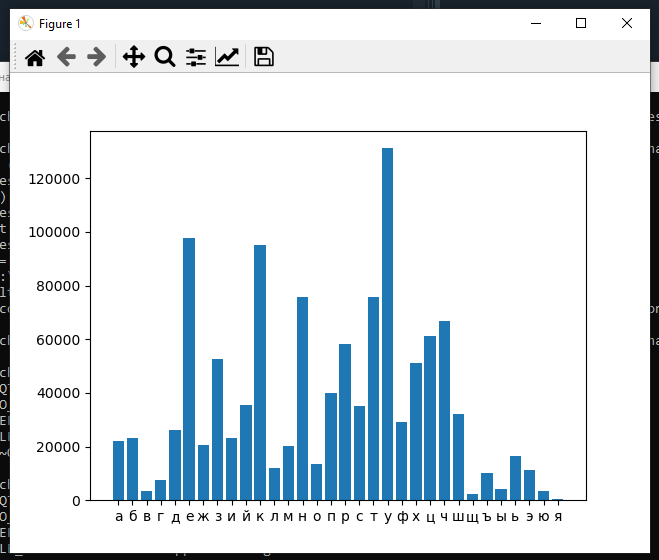


Рисунок 7 – доля символов в зашифрованном файле

Видим, что вторая гистограмма смещена относительно первой вправо на пять позиций, то есть на величину ключа.

**ВЫВОДЫ**

В результате выполнения данной лабораторной работы был изучен алгоритм работы шифра Цезаря. Была разработана программа, позволяющая зашифровать и расшифровать текст, содержащийся в файле, и представить результат работы в другой файл. Обработке подвергаются буквы латинского и русского алфавита независимо от регистра. Тестирование программы показало, что она работает правильно, расшифрованный текст совпадает с исходным. Криптоанализ шифра выявил его низкую стойкость. Зная только зашифрованный текст, можно за 32 подхода (для русского текста) перебором найти то значение ключа, при котором получится осмысленный текст. Для проверки можно вычислять долю символов зашифрованного и предполагаемого исходного текста. Она будет совпадать в паре между буквами на величину сдвига (ключа).