Министерство образования и науки Российской Федерации

1. Санкт-Петербургский Политехнический Университет Петра Великого
2. —
3. Институт компьютерных наук и кибербезопасности

**ЛАБОРАТОРНАЯ РАБОТА № 2**

1. «**Основы частотного криптоанализа**»
2. по дисциплине «Основы информационной безопасности»
3. Выполнил Кириллов Д.А
4. студент гр.

<*подпись*>

1. Преподаватель Орел Е.В

<*подпись*>

1. Санкт-Петербург
2. 2025

**Цель работы**

Приобретение навыков криптоанализа, ознакомление со способом дешифрования криптограмм на примере применения метода частотного криптоанализа.

**Постановка задачи**

Необходимо написать программу, реализующую следующие функции:

* анализ частоты букв во входном файле и вывод предполагаемых замен в соответствии с частотами распределения букв русского алфавита;
* вывод на экран всех слов, сгруппированных по количеству букв;
* вывод на экран всех слов, сгруппированных по количеству нерасшифрованных на данный момент букв;
* отображение криптограммы с указанием расшифрованного на данный момент текста и букв, которые были заменены;
* возможность замены букв в криптограмме;
* хранение и откат истории замены букв в криптограмме;
* интерфейс взаимодействия с пользователем.

Также необходимо доработать программу, реализовав функцию автоматической замены букв.

**Тестирование и результаты работы программы**

**Исходная криптограмма:**

щзф слзлпеал ф чзеылыфф пеыыгч ыл мезеыюфзялюэн эьчзеыыьэюх фыиьзщекфф, йьшлл юьмь сзф слзлпеал пеыыгч сь элюф жэлмпе сзфэяюэюжялю ылыяшлжьц сзьклыю ьдфйьб. нефйьшлл сзьэюьц эсьэьй сзьжлзбф клшьэюыьэюф пеыыгч, слзлпежелщгч ж кфизьжьщ сзлпэюежшлыфф, оюь щлюьп бьыюзьшхыгч эящщ зеээщьюзлыыгц ж шейьзеюьзыьц зейьюл ыьщлз юзф. зепеае сьщлчьяэюьцафжьмь бьпфзьжеыфн ыл юьшхбь ьйыезяуфюх ьдфйбф, ыь ф сьтжьшфюх фэсзежфюх фч йлт сьжюьзыьц слзлпеаф йшьбе пеыыгч. кьпфзьжеыфл э фэсзежшлыфлщ ьдфйьб сзлпэюежшнлю эьйьц щлюьп ьйзейьюбф эфмыешьж, сзлпыетыеалыыгц пшн яжлшфалыфн ыеплуыьэюф слзлпеаф сь кфизьжгщ беыешещ. хьюн зетшфаыгл эчлщг бьпфзьжеыфн ьалых ыл сьчьуф пзям ые пзяме ф ьэыьжеыг ые зетшфаыгч щеюлщеюфалэбфч юльзфнч, жэлщ фщ сзфэярф пже ьйрфч эжьцэюже. опыь фт ыфч фэсьшхтьжеыфл фтйгюьаыьэюф. зебьпфзьжеыыгл кфизьжгл эььйрлыфн жэлмпе эьплзуею пьсьшыфюлшхыгл, фшф фтйгюьаыгл, эфщжьшг. эюф эфщжьшг фэсьшхтяъю пшн юьмь, аюьйг сьпалзбыяюх фыпфжфпяешхыьэюх беупьмь эььйрлыфн. ич жэлмпе жгйфзеъю юеб, аюьйг эплшеюх щешьжлзьнюыьц сьюлзъ эььйрлыфлщ лмь фыпфжфпяешхыьэюф фт-те фэбеулыфн сзф жьтплцэюжфф сьщлч пьэюеюьаыь йьшхдьмь афэше эфщжьшьж. вюьзьл эжьцэюжь эьэюьфю ж яэзлпылыфф дяще. эиилбю яэзлпылыфн пьэюфмелюэн те эалю юьмь, аюь фтйгюьаыгл эфщжьшг тежфэню ью ылэбьшхбфч фыиьзщекфьыыгч эфщжьшьж. ффтфалэбен эзлпе, сь бьюьзьц слзлпеъюэн пеыыгл, ыл щьулю йгюх ейэьшъюыь ыеплуыьц. бьшлл юьмь, язьжлых дяще йгжелю ьалых жгэьбфщ, ыесзфщлз, ж йлэсзьжьпыгч эфэюлщеч эжнтф ф юлшлиьыыгч эфэюлщеч. одфйбф сзф слзлпеал — оюь злешхыьэюх, бьюьзяъ ыепь ьйнтеюлшхыь яафюгжеюх.

Таблица 1 - Распределение частот букв в криптограмме и предполагаемые замены

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Буква | Частота | Замена | Буква | Частота | Замена |
| ь | 0.1205 | О | а | 0.0196 | Ы |
| ф | 0.0873 | Е | х | 0.0188 | Ь |
| л | 0.0836 | А | ч | 0.0188 | Г |
| ы | 0.0715 | И | я | 0.0166 | З |
| е | 0.0670 | Н | н | 0.0158 | Б |
| ю | 0.0648 | Т | т | 0.0143 | Ч |
| э | 0.0580 | С | м | 0.0120 | Й |
| з | 0.0520 | Р | ц | 0.0105 | Х |
| ж | 0.0414 | В | и | 0.0068 | Ж |
| п | 0.0384 | Л | к | 0.0060 | Ш |
| ш | 0.0309 | К | у | 0.0060 | Ю |
| с | 0.0271 | М | д | 0.0053 | Ц |
| г | 0.0264 | Д | ъ | 0.0045 | Щ |
| щ | 0.0256 | П | р | 0.0038 | Э |
| й | 0.0233 | У | о | 0.0030 | Ф |
| б | 0.0196 | Я | в | 0.0008 | Ъ |

**Дешифрование:**

1. По статистике замена ь на О;
2. Замена Ж на в: по статистике и так как встречается в виде одиночного предлога.
3. Проанализировав текст, можно понять то ф – это союз И.
4. Замена ш на Л, т.к близко по статистике и подходит к слову ИшИ.
5. Э является союзом с из статистики.
6. Из слова СИщВОЛОВ можно понять, что щ – это М, и в статистике эта буква находится близко.
7. ВСлМ похоже на ВСЕМ, значит л – Е.
8. йОЛЕЕ – БОЛЕЕ, й – Б.
9. СсОСОБ – СПОСОБ, с – П.
10. БЕт – БЕЗ, т – З.
11. СОБОц – СОБОЙ, ц – Й.
12. ИЗЗе – ИЗЗА, е – А.
13. БЛОбА – БЛОКА, б –К.
14. ОдИБКИ – ОШИБКИ, д – Ш.
15. СВОЙСюВА – СВОЙСТВА, ю-Т.
16. ПОМЕч – ПОМЕХ, ч – Х.
17. ВгСОКИМ – ВЫСОКИМ, г – Ы.
18. СВнЗИ – СВЯЗИ, н – Я.
19. ТЕОзИЯХ – ТЕОРИЯХ. з – Р.
20. ЛАБОРАТОРыОЙ-ЛАБОРАТОРНОЙ, ы – Н,
21. НЕСКОЛхКИХ – НЕСКОЛЬКИХ, х –Ь.
22. МОуЕТ – МОЖЕТ, у – Ж.
23. АБСОЛъТНО – АБСОЛЮТНО, ъ – Ю.
24. пАННЫЕ – ДАННЫЕ, п – Д.
25. яРОВЕНЬ – УРОВЕНЬ, я – У.
26. ПРЕДНАЗНАаЕННЫЙ – ПРЕДНАЗНАЧЕННЫЙ, а – Ч.
27. БОЛЬЩОмО – БОЛЬШОГО, м-Г.
28. кЕЛОСТНОСТИ – ЦЕЛОСТНОСТИ, к – Ц.
29. ИНиОРМАЦИИ – ИНФОРМАЦИИ, и – Ф.
30. СООБрЕНИЯ – СООБЩЕНИЯ, р – Щ.
31. оТО – ЭТО, о – Э.
32. в – Ъ по статистике

Таблица 2 – Ключ к тексту

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п |
| Ч | К | Ъ | Ы | Ш | А | В | Р | Ф | Б | Ц | Е | Г | Я | Э | Д |
| р | с | т | у | ф | х | ц | ч | ш | щ | ъ | ы | ь | э | ю | я |
| Щ | П | З | Ж | И | Ь | Й | Х | Л | М | Ю | Н | О | С | Т | У |

**Дешифрованное сообщение:**

ПРИ ПЕРЕДАЧЕ И ХРАНЕНИИ ДАННЫХ НЕ ГАРАНТИРУЕТСЯ СОХРАННОСТЬ ИНФОРМАЦИИ, БОЛЕЕ ТОГО ПРИ ПЕРЕДАЧЕ ДАННЫХ ПО СЕТИ ВСЕГДА ПРИСУТСТВУЕТ НЕНУЛЕВОЙ ПРОЦЕНТ ОШИБОК. НАИБОЛЕЕ ПРОСТОЙ СПОСОБ ПРОВЕРКИ ЦЕЛОСТНОСТИ ДАННЫХ, ПЕРЕДАВАЕМЫХ В ЦИФРОВОМ ПРЕДСТАВЛЕНИИ, ЭТО МЕТОД КОНТРОЛЬНЫХ СУММ РАССМОТРЕННЫЙ В ЛАБОРАТОРНОЙ РАБОТЕ НОМЕР ТРИ. РАДАЧА ПОМЕХОУСТОЙЧИВОГО КОДИРОВАНИЯ НЕ ТОЛЬКО ОБНАРУЖИТЬ ОШИБКИ, НО И ПОЗВОЛИТЬ ИСПРАВИТЬ ИХ БЕЗ ПОВТОРНОЙ ПЕРЕДАЧИ БЛОКА ДАННЫХ. ЦОДИРОВАНИЕ С ИСПРАВЛЕНИЕМ ОШИБОК ПРЕДСТАВЛЯЕТ СОБОЙ МЕТОД ОБРАБОТКИ СИГНАЛОВ, ПРЕДНАЗНАЧЕННЫЙ ДЛЯ УВЕЛИЧЕНИЯ НАДЕЖНОСТИ ПЕРЕДАЧИ ПО ЦИФРОВЫМ КАНАЛАМ. ХОТЯ РАЗЛИЧНЫЕ СХЕМЫ КОДИРОВАНИЯ ОЧЕНЬ НЕ ПОХОЖИ ДРУГ НА ДРУГА И ОСНОВАНЫ НА РАЗЛИЧНЫХ МАТЕМАТИЧЕСКИХ ТЕОРИЯХ, ВСЕМ ИМ ПРИСУЩИ ДВА ОБЩИХ СВОЙСТВА. ЭДНО ИЗ НИХ ИСПОЛЬЗОВАНИЕ ИЗБЫТОЧНОСТИ. РАКОДИРОВАННЫЕ ЦИФРОВЫЕ СООБЩЕНИЯ ВСЕГДА СОДЕРЖАТ ДОПОЛНИТЕЛЬНЫЕ, ИЛИ ИЗБЫТОЧНЫЕ, СИМВОЛЫ. СТИ СИМВОЛЫ ИСПОЛЬЗУЮТ ДЛЯ ТОГО, ЧТОБЫ ПОДЧЕРКНУТЬ ИНДИВИДУАЛЬНОСТЬ КАЖДОГО СООБЩЕНИЯ. ФХ ВСЕГДА ВЫБИРАЮТ ТАК, ЧТОБЫ СДЕЛАТЬ МАЛОВЕРОЯТНОЙ ПОТЕРЮ СООБЩЕНИЕМ ЕГО ИНДИВИДУАЛЬНОСТИ ИЗ-ЗА ИСКАЖЕНИЯ ПРИ ВОЗДЕЙСТВИИ ПОМЕХ ДОСТАТОЧНО БОЛЬШОГО ЧИСЛА СИМВОЛОВ. ВТОРОЕ СВОЙСТВО СОСТОИТ В УСРЕДНЕНИИ ШУМА. СФФЕКТ УСРЕДНЕНИЯ ДОСТИГАЕТСЯ ЗА СЧЕТ ТОГО, ЧТО ИЗБЫТОЧНЫЕ СИМВОЛЫ ЗАВИСЯТ ОТ НЕСКОЛЬКИХ ИНФОРМАЦИОННЫХ СИМВОЛОВ. ИИЗИЧЕСКАЯ СРЕДА, ПО КОТОРОЙ ПЕРЕДАЮТСЯ ДАННЫЕ, НЕ МОЖЕТ БЫТЬ АБСОЛЮТНО НАДЕЖНОЙ. КОЛЕЕ ТОГО, УРОВЕНЬ ШУМА БЫВАЕТ ОЧЕНЬ ВЫСОКИМ, НАПРИМЕР, В БЕСПРОВОДНЫХ СИСТЕМАХ СВЯЗИ И ТЕЛЕФОННЫХ СИСТЕМАХ. ЭШИБКИ ПРИ ПЕРЕДАЧЕ — ЭТО РЕАЛЬНОСТЬ, КОТОРУЮ НАДО ОБЯЗАТЕЛЬНО УЧИТЫВАТЬ.

**Результат автоматической замены:**

ПЗФ СЛЗЛПЕАЛ Ф ЧЗЕЫЛЫФФ ПЕЫЫГЧ ЫЛ МЕЗЕЫЮФЗЯЛЮЭН ЭЬЧЗЕЫЫЬЭЮХ ФЫИЬЗЩЕКФФ, ЙЬШЛЛ ЮЬМЬ СЗФ СЛЗЛПЕАЛ ПЕЫЫГЧ СЬ ЭЛЮФ ЖЭЛМПЕ СЗФЭЯЮЭЮЖЯЛЮ ЫЛЫЯШЛЖЬЦ СЗЬКЛЫЮ ЬДФЙЬБ. НЕФЙЬШЛЛ СЗЬЭЮЬЦ ЭСЬЭЬЙ СЗЬЖЛЗБФ КЛШЬЭЮЫЬЭЮФ ПЕЫЫГЧ, СЛЗЛПЕЖЕЛЩГЧ Ж КФИЗЬЖЬЩ СЗЛПЭЮЕЖШЛЫФФ, ОЮЬ ЩЛЮЬП БЬЫЮЗЬШХЫГЧ ЭЯЩЩ ЗЕЭЭЩЬЮЗЛЫЫГЦ Ж ШЕЙЬЗЕЮЬЗЫЬЦ ЗЕЙЬЮЛ ЫЬЩЛЗ ЮЗФ. ЗЕПЕАЕ СЬЩЛЧЬЯЭЮЬЦАФЖЬМЬ БЬПФЗЬЖЕЫФН ЫЛ ЮЬШХБЬ ЬЙЫЕЗЯУФЮХ ЬДФЙБФ, ЫЬ Ф СЬТЖЬШФЮХ ФЭСЗЕЖФЮХ ФЧ ЙЛТ СЬЖЮЬЗЫЬЦ СЛЗЛПЕАФ ЙШЬБЕ ПЕЫЫГЧ. КЬПФЗЬЖЕЫФЛ Э ФЭСЗЕЖШЛЫФЛЩ ЬДФЙЬБ СЗЛПЭЮЕЖШНЛЮ ЭЬЙЬЦ ЩЛЮЬП ЬЙЗЕЙЬЮБФ ЭФМЫЕШЬЖ, СЗЛПЫЕТЫЕАЛЫЫГЦ ПШН ЯЖЛШФАЛЫФН ЫЕПЛУЫЬЭЮФ СЛЗЛПЕАФ СЬ КФИЗЬЖГЩ БЕЫЕШЕЩ. ХЬЮН ЗЕТШФАЫГЛ ЭЧЛЩГ БЬПФЗЬЖЕЫФН ЬАЛЫХ ЫЛ СЬЧЬУФ ПЗЯМ ЫЕ ПЗЯМЕ Ф ЬЭЫЬЖЕЫГ ЫЕ ЗЕТШФАЫГЧ ЩЕЮЛЩЕЮФАЛЭБФЧ ЮЛЬЗФНЧ, ЖЭЛЩ ФЩ СЗФЭЯРФ ПЖЕ ЬЙРФЧ ЭЖЬЦЭЮЖЕ. ОПЫЬ ФТ ЫФЧ ФЭСЬШХТЬЖЕЫФЛ ФТЙГЮЬАЫЬЭЮФ. ЗЕБЬПФЗЬЖЕЫЫГЛ КФИЗЬЖГЛ ЭЬЬЙРЛЫФН ЖЭЛМПЕ ЭЬПЛЗУЕЮ ПЬСЬШЫФЮЛШХЫГЛ, ФШФ ФТЙГЮЬАЫГЛ, ЭФЩЖЬШГ. ЭЮФ ЭФЩЖЬШГ ФЭСЬШХТЯЪЮ ПШН ЮЬМЬ, АЮЬЙГ СЬПАЛЗБЫЯЮХ ФЫПФЖФПЯЕШХЫЬЭЮХ БЕУПЬМЬ ЭЬЬЙРЛЫФН. ИЧ ЖЭЛМПЕ ЖГЙФЗЕЪЮ ЮЕБ, АЮЬЙГ ЭПЛШЕЮХ ЩЕШЬЖЛЗЬНЮЫЬЦ СЬЮЛЗЪ ЭЬЬЙРЛЫФЛЩ ЛМЬ ФЫПФЖФПЯЕШХЫЬЭЮФ ФТ-ТЕ ФЭБЕУЛЫФН СЗФ ЖЬТПЛЦЭЮЖФФ СЬЩЛЧ ПЬЭЮЕЮЬАЫЬ ЙЬШХДЬМЬ АФЭШЕ ЭФЩЖЬШЬЖ. ВЮЬЗЬЛ ЭЖЬЦЭЮЖЬ ЭЬЭЮЬФЮ Ж ЯЭЗЛПЫЛЫФФ ДЯЩЕ. ЭИИЛБЮ ЯЭЗЛПЫЛЫФН ПЬЭЮФМЕЛЮЭН ТЕ ЭАЛЮ ЮЬМЬ, АЮЬ ФТЙГЮЬАЫГЛ ЭФЩЖЬШГ ТЕЖФЭНЮ ЬЮ ЫЛЭБЬШХБФЧ ФЫИЬЗЩЕКФЬЫЫГЧ ЭФЩЖЬШЬЖ. ФФТФАЛЭБЕН ЭЗЛПЕ, СЬ БЬЮЬЗЬЦ СЛЗЛПЕЪЮЭН ПЕЫЫГЛ, ЫЛ ЩЬУЛЮ ЙГЮХ ЕЙЭЬШЪЮЫЬ ЫЕПЛУЫЬЦ. БЬШЛЛ ЮЬМЬ, ЯЗЬЖЛЫХ ДЯЩЕ ЙГЖЕЛЮ ЬАЛЫХ ЖГЭЬБФЩ, ЫЕСЗФЩЛЗ, Ж ЙЛЭСЗЬЖЬПЫГЧ ЭФЭЮЛЩЕЧ ЭЖНТФ Ф ЮЛШЛИЬЫЫГЧ ЭФЭЮЛЩЕЧ. ОДФЙБФ СЗФ СЛЗЛПЕАЛ — ОЮЬ ЗЛЕШХЫЬЭЮХ, БЬЮЬЗЯЪ ЫЕПЬ ЬЙНТЕЮЛШХЫЬ ЯАФЮГЖЕЮХ.

**Контрольные вопросы:**

1) Шифр моноалфавитной подстановки – шифр, при котором каждый символ открытого текста заменяется на некоторый, фиксированный при данном ключе символ того же алфавита.

2) Главным недостатком шифра моноалфавитной подстановки является то, что его легко вычислить и получить открытый текст без ключа при помощи частотного криптоанализа.

3) Сложность для дешифрации методом прямого перебора сообщения, зашифрованного шифром моноалфавитной подстановки – О (33!) для русского языка и О(N!) для любого языка, где N – мощность алфавита.

4) Упростить частотный анализ может длинный текст, так как в таком случае частоты отдельных букв будут ближе к табличным значениям; наличие коротких слов (1–3 символа), по которым расшифровываются первые буквы и с которых начинается расшифровка. Кроме того, частотность букв зависит и от рода текста, поэтому для более точного определения необходимо проводить аналитику по нескольким текстам и расшифровывать их.

5) Чем больше текст, тем ближе будут частоты букв к табличным значениям, и при дешифровании первым способом (то есть заменяя буквы исходя из статистики) текст будет уже близок к полностью дешифрованному. Но чем меньше текст, тем хуже будет проявляться статистика. Следовательно, после автоматической замены букв в небольшом тексте, расшифрованный текст будет мало похож на полностью дешифрованный.

**Вывод**

В ходе лабораторной работы был изучен шифр моноалфавитной подстановки, а также методы частотного криптоанализа, применимые для его расшифровки. Так как был предоставлен небольшой текст размером около 1500 символов, после автоматической замены не был получен полностью расшифрованный текст или хотя бы отдаленно похожий на него. В основном частотное распределение помогло сделать первые шаги для получения более-менее узнаваемых слов, далее производилась ручная замена букв. В результате лабораторной работы стало ясно, что моноалфавитный шифр является небезопасным, так как даже в небольшом тексте, алгоритм смог правильно заменить 6 букв, а остальные не так сложно поддались ручной расшифровке.

**ПРИЛОЖЕНИЕ А**

# Lab2

# Функция, анализирующая шифрограмму и возвращающая списки отсортированных частот каждых букв в шифрограмме и русском алфавите в виде (буква, частота)

def analises(my\_cryptogram):

    rus\_letter\_freq = {

    'а': 0.0801, 'б': 0.0159, 'в': 0.0454, 'г': 0.0170, 'д': 0.0298,

    'е': 0.0845, 'ё': 0.0003, 'ж': 0.0094, 'з': 0.0165, 'и': 0.0735,

    'й': 0.0121, 'к': 0.0349, 'л': 0.0440, 'м': 0.0321, 'н': 0.0670,

    'о': 0.1097, 'п': 0.0281, 'р': 0.0473, 'с': 0.0547, 'т': 0.0626,

    'у': 0.0262, 'ф': 0.0026, 'х': 0.0097, 'ц': 0.0048, 'ч': 0.0144,

    'ш': 0.0073, 'щ': 0.0036, 'ъ': 0.0004, 'ы': 0.0190, 'ь': 0.0174,

    'э': 0.0032, 'ю': 0.0064, 'я': 0.0201

    }

    sorted\_freq = sorted(rus\_letter\_freq.items(), key = lambda x: -x[1])

    # Генератор, который очищает текст от небуквенных символов

    clean\_my\_cryptogram = ''.join([char for char in my\_cryptogram if char.isalpha()])

    cryptogram\_letter\_freq = {}

    len\_cryptogram = len(clean\_my\_cryptogram)

    # цикл, добавляющий в список букву и ее количество в шифрограмме

    for letter in clean\_my\_cryptogram:

        if letter in cryptogram\_letter\_freq:

            cryptogram\_letter\_freq[letter] += 1

        else:

            cryptogram\_letter\_freq[letter] = 1

    for letter in cryptogram\_letter\_freq:

        cryptogram\_letter\_freq[letter] = round(cryptogram\_letter\_freq[letter] / len\_cryptogram, 4)

    sorted\_my\_freq = sorted(cryptogram\_letter\_freq.items(), key = lambda x: -x[1])

    return sorted\_my\_freq, sorted\_freq

# Функция, выводящая на экран слова, отсортированные по количеству букв

def print\_group(my\_cryptogram):

    # Разделение текста на слова и удаление небуквенных символов

    words = my\_cryptogram.split()

    delited\_waste = []

    for word in words:

        del\_waste = ''.join([char for char in word if char.isalpha()])

        delited\_waste.append(del\_waste)

    len\_matrix = max(len(word) for word in delited\_waste)

    matrix = [[] for \_ in range(len\_matrix+1)]

    # Заполняем двумерный список словами и после выводим его

    for word in delited\_waste:

        matrix[len(word)].append(word)

    for i in range(1,len\_matrix+1):

        if matrix[i]:

            print(f"слова с {i} букв", matrix[i])

# Функция, которая выводит на экран слова, отсортированные по количеству нерасшифрованных букв

def print\_decrypted(my\_cryptogram, decr\_letters):

    # Разделение текста на слова и удаление небуквенных символов

    words = my\_cryptogram.split()

    delited\_waste = []

    for word in words:

        del\_waste = ''.join([char for char in word if char.isalpha()])

        delited\_waste.append(del\_waste)

    # Длелаем множество из букв, которые нерасшифрованы

    letters = [char for char in my\_cryptogram if char.isalpha()]

    decr\_letters = list(decr\_letters)

    letters = [char for char in letters if char not in decr\_letters]

    # Создаем двумерный массив и заполняем его словами по порядку нерасшифрованных букв и выводим его

    len\_letters = max(len(x) for x in delited\_waste)

    matrix = [[] for \_ in range(len\_letters+1)]

    for word in delited\_waste:

        cnt = 0

        for char in word:

            if char in letters:

                cnt +=1

        matrix[cnt].append(word)

    for i in range(len\_letters,0, -1):

        if matrix[i]:

            print(f"нерасшифрованно {i} букв: ", matrix[i])

def main():

    with open("cryptogram.txt", 'r', encoding='utf-8') as f:

        original = f.read()

    newtext = original

    history = []

    substitutions = {}

    while(True):

        print("Выберите действие:\n\

            1.Анализ частоты и вывод предполагаемых замен.\n\

            2.Вывод всех слов, сгруппирированных по количеству букв.\n\

            3.Вывод всех слов, сгруппирированных по количестку нерасшифрованных букв.\n\

            4.Замена букв в криптограмме\n\

            5.Откат истории.\n\

            6.Автоматическая замена букв.\n\

            7.Вывод текущего текста и замен.\n\

            0.Выход из программы")

        action = int(input())

        if action not in [1,2,3,4,5,6,7,0]:

            print("Нет такого действия")

        match action:

            case 1:

                my\_freq, rus\_lett\_freq = analises(original)

                min\_len = min(len(my\_freq),len(rus\_lett\_freq))

                print("Предполагаемые замены:\n")

                for i in range(min\_len):

                    print(f"{my\_freq[i][0]} -> {rus\_lett\_freq[i][0]}  {my\_freq[i][1]}  {rus\_lett\_freq[i][1]}")

                print(f"Необработанные буквы {rus\_lett\_freq[len(my\_freq):]}")

            case 2:

                print\_group(newtext)

            case 3:

                decrypted\_letters = substitutions.values()

                print\_decrypted(newtext, decrypted\_letters)

            case 4:

                letter1, letter2 = input(f"Заменить букву X на Y (введите через пробел): ").split()

                alphabet = ['а', 'б', 'в', 'г', 'д', 'е', 'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ъ', 'ы', 'ь', 'э', 'ю', 'я']

                if (letter1 not in alphabet or letter2 not in alphabet):

                    print("Нет такой буквы в алфавите")

                    continue

                substitutions[letter1] = letter2.upper()

                history.append((letter1, letter2.upper()))

                newtext = ''.join([substitutions.get(char, char) for char in original])

            case 5:

                n = len(history)

                steps = int(input(f"Введите на сколько шагов вернуть назад (вы можете вернуться не более, чем на {n} шагов): "))

                if steps > n:

                    print("Нельзя вернуться на столько шагов")

                    continue

                for \_ in range(steps):

                    old\_char, \_ = history.pop()

                    del substitutions[old\_char]

                temp\_text = original

                for enc, dec in substitutions.items():

                    temp\_text = temp\_text.replace(enc, dec)

                newtext = temp\_text

            case 6:

                    my\_freq, rus\_freq = analises(original)

                    for i in range(min(len(my\_freq), len(rus\_freq))):

                        substitutions[my\_freq[i][0]] = rus\_freq[i][0]

                        history.append((my\_freq[i][0], rus\_freq[i][0]))

                    temp\_text = original

                    for enc, dec in substitutions.items():

                        temp\_text = temp\_text.replace(enc, dec)

                    newtext = newtext = ''.join([substitutions.get(char.upper(), char.upper()) for char in original])

                    print(newtext)

            case 7:

                print(f"Текущий текст:\n\n{newtext}")

                print("Текущие замены:\n\n")

                for old, new in substitutions.items():

                    print(old,"->",new)

            case 0:

                break

main()