МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

**ЛАБОРАТОРНЫЕ РАБОТЫ ПО ПРЕДМЕТУ**

«Программирование криптографических алгоритмов»

**Выполнил:**

Барышников С.С. гр. 191-351

**Преподаватель:**

Бутакова Н.Г.

Москва 2021 г.

**Содержание**

[Аннотация 3](#_Toc83852118)

[Постоянный модуль 4](#_Toc83852119)

[Блок D: ШИФРЫ ПЕРЕСТАНОВКИ……………………………………………](#_Toc83852120)

[5](#_Toc83852120)

[10. Шифр вертикальной перестановки 5](#_Toc83852130)

[11. Решетка Кардано 9](#_Toc83852131)

# Аннотация

**Среда программирования:** Visual Studio Code

**Язык программирования:** Python 3

**Процедуры для запуска программы:** $ python3 <имя\_файла>.py

**Пословица-тест:** Время, приливы и отливы не ждут человека.

**Текст для проверки работы:** Вот пример статьи на тысячу символов. Это достаточно маленький текст, оптимально подходящий для карточек товаров в интернет или магазинах или для небольших информационных публикаций. В таком тексте редко бывает более двух или трёх абзацев и обычно один подзаголовок. Но можно и без него. На тысячу символов рекомендовано использовать один или два ключа и одну картину. Текст на тысячу символов это сколько примерно слов? Статистика показывает, что тысяча включает в себя сто пятьдесят или двести слов средней величины. Но, если злоупотреблять предлогами, союзами и другими частями речи на один или два символа, то количество слов неизменно возрастает. В копирайтерской деятельности принято считать тысячи с пробелами или без. Учет пробелов увеличивает объем текста примерно на сто или двести символов именно столько раз мы разделяем слова свободным пространством. Считать пробелы заказчики не любят, так как это пустое место. Однако некоторые фирмы и биржи видят справедливым ставить стоимость за тысячу символов с пробелами, считая последние важным элементом качественного восприятия. Согласитесь, читать слитный текст без единого пропуска, никто не будет. Но большинству нужна цена за тысячу знаков без пробелов.

**Интерфейс:** #в разработке#

# Постоянный модуль

Код модуля base.py используемый для предотвращения дублирования кода, используется во всех последующих программах:

import re

alphabet = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя"

dict = {'.': 'тчк', ',': 'зпт'}

def replace\_all\_to(input\_text, dict):

    input\_text = input\_text.replace(' ', '')

    for i, j in dict.items():

        input\_text = input\_text.replace(i, j)

    return input\_text

def replace\_all\_from(input\_text, dict):

    for i, j in dict.items():

        input\_text = input\_text.replace(j, i)

    return input\_text

def file\_to\_string(name):

    with open(name) as f:

        input\_short\_text = " ".join([l.rstrip() for l in f]) + ' '

    return input\_short\_text.lower()

def input\_for\_cipher\_short():

    return replace\_all\_to(file\_to\_string('short.txt'), dict)

def input\_for\_cipher\_long():

    return replace\_all\_to(file\_to\_string('long.txt'), dict)

def output\_from\_decrypted(decrypted\_text):

    return replace\_all\_from(decrypted\_text, dict)

**D: ШИФРЫ ПЕРЕСТАНОВКИ**



## 

## Шифр вертикальной перестановки

Широкое распространение получила разновидность маршрутной перестановки — вертикальная перестановка. В этом шифре также используется прямоугольная таблица, в которую сообщение записывается по строкам слева направо. Выписывается шифрограмма по вертикалям, при этом столбцы выбираются в порядке, определяемом ключом.

**Код программы:**

from base import alphabet, input\_for\_cipher\_short, input\_for\_cipher\_long, output\_from\_decrypted

import math

key = str(input('Введите ключ: '))

def transposition\_encode(msg, key):

    cipher = ""

    k\_indx = 0

    msg\_len = float(len(msg))

    msg\_lst = list(msg)

    key\_lst = sorted(list(key))

    col = len(key)

    row = int(math.ceil(msg\_len / col))

    fill\_null = int((row \* col) - msg\_len)

    msg\_lst.extend('\_' \* fill\_null)

    matrix = [msg\_lst[i: i + col] for i in range(0, len(msg\_lst), col)]

    for \_ in range(col):

        curr\_idx = key.index(key\_lst[k\_indx])

        cipher += ''.join([row[curr\_idx] for row in matrix])

        k\_indx += 1

    return cipher

def transposition\_decode(cipher, key):

    msg = ""

    k\_indx = 0

    msg\_indx = 0

    msg\_len = float(len(cipher))

    msg\_lst = list(cipher)

    col = len(key)

    row = int(math.ceil(msg\_len / col))

    key\_lst = sorted(list(key))

    dec\_cipher = []

    for \_ in range(row):

        dec\_cipher += [[None] \* col]

    for \_ in range(col):

        curr\_idx = key.index(key\_lst[k\_indx])

        for j in range(row):

            dec\_cipher[j][curr\_idx] = msg\_lst[msg\_indx]

            msg\_indx += 1

        k\_indx += 1

    null\_count = msg.count('\_')

    if null\_count > 0:

        return msg[: -null\_count]

    msg = ''.join(sum(dec\_cipher, []))

    return msg.replace('\_', '')

# вывод результатов работы программы

print(f'''

Шифр вертикальной перестановки:

КОРОТКИЙ ТЕКСТ:

Зашифрованный текст:

{transposition\_encode(input\_for\_cipher\_short(), key)}

Расшифрованный текст:

{output\_from\_decrypted(transposition\_decode(transposition\_encode(

    input\_for\_cipher\_short(), key), key))}

ДЛИННЫЙ ТЕКСТ:

Зашифрованный текст:

{transposition\_encode(input\_for\_cipher\_long(), key)}

Расшифрованный текст:

{output\_from\_decrypted(transposition\_decode(transposition\_encode(

    input\_for\_cipher\_long(), key), key))}

''')

**Тестирование:**

/bin/python3 /root/mospolytech-education-crypt-dev-2021-1/lab04\_10\_transposition.py

Введите ключ: ключ

Шифр вертикальной перестановки:

КОРОТКИЙ ТЕКСТ:

Зашифрованный текст:

вяпиовжчвтрзрвтыдеечмтлииетоа\_епиылнулкк

Расшифрованный текст:

время,приливыиотливынеждутчеловека.

ДЛИННЫЙ ТЕКСТ:

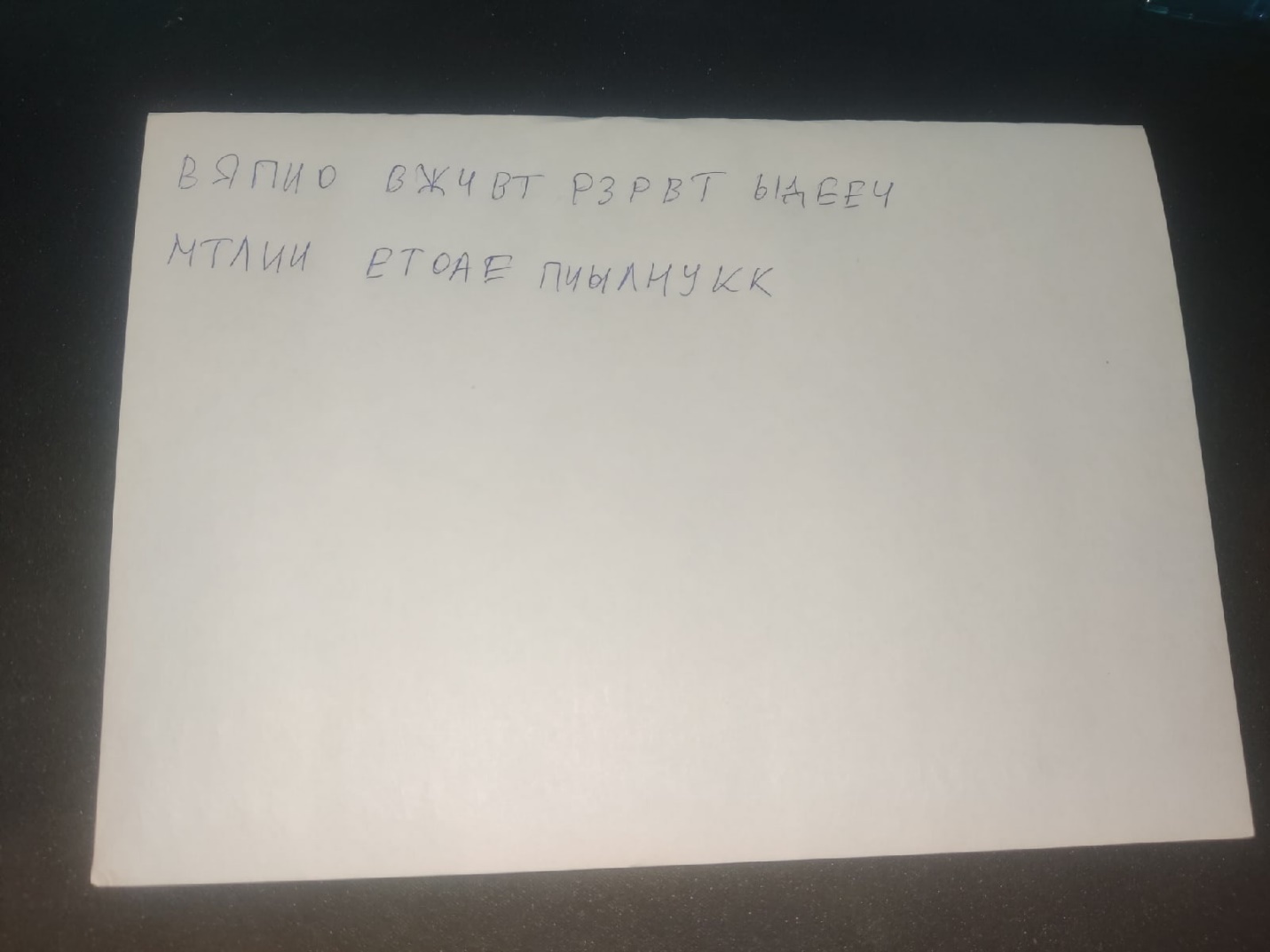
Зашифрованный текст:

врртаяилчотчаьттомндядачооннлгнляоифанпииккееквбехтацонидоокоинтаяилеевилвоивюокичкаяилтоомовсикквзтсвчвяпдтдтоейиынтиуряроисадиамчоивматлсснморачоарйтнинсаыиоалзупеуиабттинслесоиноомзялсомсноктпеачнбпкэуеткаеофырисвиситозссосбмттодвыетатнвриклтзиьтттеорспкеекоивжеаянвпетоисьтчмокданлкезпаохщлреввтеиааинлхоцнукйвокроаодирбебонзлкнжбечтчмокнасьадлачданкстчмоолпесттсааапоякассяеививдвчтоезпетегзомрмсиидлавзоитлееваекпйсдеопячтссбмитчрлвчеъеамотисилмолрыдеовдптсмсарлкиеятктсмооккрииждпевттосаяилпеисаснаммочвооиясаептснебдгокттбтнлнуннтчабрлчпеанссоттсомнйстиьодйкотририаиилбшнмохлцчаттдытеуихаичдогвчмозонссормооооьндлиуттенссоэккиноктиоытчыаютбоьяислрелнкплотлплмтзигчяеандилтоеовзнзтткреояьтиоттчрлиектбвлвомсрраививвнткзалсабыоавчиьбззиюзакпосчннтемивталмвсмьыуввоапипеенлнксноптчгиьчтийсзнпузиндчбштуцзсзозбв\_тмтиыуввэотоеикптлпоияткаветмзхдеьириыбаттмсебелвлёзвыопаотонегкыувводнпзтииканруттыуввсьррлчатпзеттчлееттслесснеичзслобьдапюиуитрниисопкчвоиноствиткелсртиьяпеибчеооеитекпенодтмоесьаремвонррттчтоыаклттаотетдооырбиярдыаьиттчморлзчялижэемеегсятоссталыкеиопаноутоьснааыукеоок

Расшифрованный текст:

вотпримерстатьинатысячусимволов.этодостаточномаленькийтекст,оптимальноподходящийдлякарточектовароввинтернетилимагазинахилидлянебольшихинформационныхпубликаций.втакомтекстередкобываетболеедвухилитрёхабзацевиобычноодинподзаголовок.номожноибезнего.натысячусимволоврекомендованоиспользоватьодинилидваключаиоднукартину.текстнатысячусимволовэтосколькопримернослов.статистикапоказывает,чтотысячавключаетвсебястопятьдесятилидвестисловсреднейвеличины.но,еслизлоупотреблятьпредлогами,союзамиидругимичастямиречинаодинилидвасимвола,токоличествословнеизменновозрастает.вкопирайтерскойдеятельностипринятосчитатьтысячиспробеламиилибез.учетпробеловувеличиваетобъемтекстапримернонастоилидвестисимволовименностолькоразмыразделяемсловасвободнымпространством.считатьпробелызаказчикинелюбят,таккакэтопустоеместо.однаконекоторыефирмыибирживидятсправедливымставитьстоимостьзатысячусимволовспробелами,считаяпоследниеважнымэлементомкачественноговосприятия.согласитесь,читатьслитныйтекстбезединогопропуска,никтонебудет.нобольшинствунужнаценазатысячузнаковбезпробелов.

**Карточка:**



## Решетка Кардано

Решётка Кардано — исторически первая известная шифровальная решётка, трафарет, применявшийся для шифрования и дешифрования, выполненный в форме прямоугольной (чаще всего — квадратной) таблицы-карточки, часть ячеек которых вырезана, и через которые наносился шифротекст. Пустые поля текста заполнялись другим текстом для маскировки сообщений под обычные послания — таким образом, применение решётки является одной из форм стеганографии.

**Код программы:**

from base import alphabet, input\_for\_cipher\_short, input\_for\_cipher\_long, output\_from\_decrypted

class Cardan(object):

    def \_\_init\_\_(self, size, spaces):

        self.size = int(size)

        str1 = ''

        for i in range(len(spaces)):

            str1 = str1 + str(spaces[i][0]) + str(spaces[i][1])

        self.spaces = str1

        matrix\_spaces = []

        i = 0

        cont = 0

        while i < self.size\*self.size//4:

            t = int(self.spaces[cont]), int(self.spaces[cont + 1])

            cont = cont + 2

            i = i+1

            matrix\_spaces.append(t)

        self.spaces = matrix\_spaces

    def code(self, message):

        offset = 0

        cipher\_text = ""

        matrix = []

        for i in range(self.size\*2-1):

            matrix.append([])

            for j in range(self.size):

                matrix[i].append(None)

        whitesneeded = self.size\*self.size - \

            len(message) % (self.size\*self.size)

        if (len(message) % (self.size\*self.size) != 0):

            for h in range(whitesneeded):

                message = message + ' '

        while offset < len(message):

            self.spaces.sort()

            for i in range(int(self.size\*self.size//4)):

                xy = self.spaces[i]

                x = xy[0]

                y = xy[1]

                matrix[x][y] = message[offset]

                offset = offset + 1

            if (offset % (self.size\*self.size)) == 0:

                for i in range(self.size):

                    for j in range(self.size):

                        try:

                            cipher\_text = cipher\_text + matrix[i][j]

                        except:

                            pass

            for i in range(self.size\*self.size//4):

                x = (self.size-1)-self.spaces[i][1]

                y = self.spaces[i][0]

                self.spaces[i] = x, y

        return cipher\_text

    def decode(self, message, size):

        uncipher\_text = ""

        offset = 0

        matrix = []

        for i in range(self.size\*2-1):

            matrix.append([])

            for j in range(self.size):

                matrix[i].append(None)

        whitesneeded = self.size\*self.size - \

            len(message) % (self.size\*self.size)

        if (len(message) % (self.size\*self.size) != 0):

            for h in range(whitesneeded):

                message = message + ' '

        offsetmsg = len(message) - 1

        while offset < len(message):

            if (offset % (self.size\*self.size)) == 0:

                for i in reversed(list(range(self.size))):

                    for j in reversed(list(range(self.size))):

                        matrix[i][j] = message[offsetmsg]

                        offsetmsg = offsetmsg - 1

            for i in reversed(list(range(self.size\*self.size//4))):

                x = self.spaces[i][1]

                y = (self.size-1)-self.spaces[i][0]

                self.spaces[i] = x, y

            self.spaces.sort(reverse=True)

            for i in range(self.size\*self.size//4):

                xy = self.spaces[i]

                x = xy[0]

                y = xy[1]

                uncipher\_text = matrix[x][y] + uncipher\_text

                offset = offset + 1

        return uncipher\_text

gaps = [(7, 7), (6, 0), (5, 0), (4, 0), (7, 1), (1, 1), (1, 2), (4, 1), (7, 2), (2, 1), (2, 5), (2, 3), (7, 3), (3, 1), (3, 2), (3, 4)]

r = Cardan(8, gaps)

texto = input\_for\_cipher\_short()

n = len(texto)

encoded = r.code(texto)

decoded = r.decode(encoded, n)

gaps2 = [(7, 7), (6, 0), (5, 0), (4, 0), (7, 1), (1, 1), (1, 2), (4, 1), (7, 2), (2, 1), (2, 5), (2, 3), (7, 3), (3, 1), (3, 2), (3, 4)]

r2 = Cardan(8, gaps)

texto\_long = input\_for\_cipher\_long()

n = len(texto\_long)

encoded\_long = r2.code(texto\_long)

decoded\_long = r2.decode(encoded\_long, n)

print(f'''

Шифр вертикальной перестановки:

КОРОТКИЙ ТЕКСТ:

Зашифрованный текст:

{encoded.replace(' ', '')}

Расшифрованный текст:

{output\_from\_decrypted(decoded)}

ДЛИННЫЙ ТЕКСТ:

Зашифрованный текст:

{encoded\_long}

Расшифрованный текст:

{output\_from\_decrypted(decoded\_long)}

''')

**Тестирование:**

/bin/python3 /root/mospolytech-education-crypt-dev-2021-1/lab04\_11\_cardan.py

Шифр вертикальной перестановки:

КОРОТКИЙ ТЕКСТ:

Зашифрованный текст:

векаовртетмячзплткпривиынелждутчивыелои

Расшифрованный текст:

время,приливыиотливынеждутчеловека.

ДЛИННЫЙ ТЕКСТ:

Зашифрованный текст:

чалекэтанвоькийотттперкдсимыетосрсстзатятчоучпнсаимвоомлттьиовтнолимваряаопгазионтщиамхвиалиьлвинойнитедплряндекоартотичлдхоектдиерейтчадянкобыквецбаоевтльишбтаихокоомнинтыелкхнпублстиефоркацмдожнзагцоедибезонвеуехглоилвитовтриочктоёбчыккнчхноодоминабзнпоалвакьзоелатючаивоыксдянаучуосктьиммоадиевнндирлоованоидитловспоровтчлькякинстатоиучтсчтпиктуекрикссмаеритмнвопсонловэлотоатыосксядвестосскатислпозяыввсяраечтетьзпаддесвткялтниюччаетливетотсебыясоютьптзйвамиирдеелриуегчиснидлытломгаичзмлиизокупотптрчнозеблптнеиокоизасменнлотлявмоизириерчечидсатввнаоссслиамволоватодизптниысяприрчаеиспрнотстбчеялквккатоопосмчийидтеаитяртелььтниайтостебнонъемеалистоитлблеиздевтчокексучвтсапуевреитмлтичивераипроетобямпремсоосистралнмлвсотовлоьвоваимксмвооербаотдзнмыранызчносделтпеметтааскстотчккчкиотдкнатаьаакпрзэкточоипкуосибнелютобнелыятззиоствымрьекзатыссожтяочтурыиесавфивиитьирдсятмотмспраимввыибедливачеажнтсолтвеныносвосгмопрчовэлбеиеометланятсопаослемкдпмизниептогоныйтприропутсяеткиаезятсчпксксьттбезопзтендчгитатиньиласслииевбеназозктпробаеолнлеотвбуьдтысетшяччуитнзснкатчвунукож кнонацб

Расшифрованный текст:

вотпримерстатьинатысячусимволов.этодостаточномаленькийтекст,оптимальноподходящийдлякарточектовароввинтернетилимагазинахилидлянебольшихинформационныхпубликаций.втакомтекстередкобываетболеедвухилитрёхабзацевиобычноодинподзаголовок.номожноибезнего.натысячусимволоврекомендованоиспользоватьодинилидваключаиоднукартину.текстнатысячусимволовэтосколькопримернослов.статистикапоказывает,чтотысячавключаетвсебястопятьдесятилидвестисловсреднейвеличины.но,еслизлоупотреблятьпредлогами,союзамиидругимичастямиречинаодинилидвасимвола,токоличествословнеизменновозрастает.вкопирайтерскойдеятельностипринятосчитатьтысячиспробеламиилибез.учетпробеловувеличиваетобъемтекстапримернонастоилидвестисимволовименностолькоразмыразделяемсловасвободнымпространством.считатьпробелызаказчикинелюбят,таккакэтопустоеместо.однаконекоторыефирмыибирживидятсправедливымставитьстоимостьзатысячусимволовспробелами,считаяпоследниеважнымэлементомкачественноговосприятия.согласитесь,читатьслитныйтекстбезединогопропуска,никтонебудет.нобольшинствунужнаценазатысячузнаковбезпробелов.

**Карточка:**

