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

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

10.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:

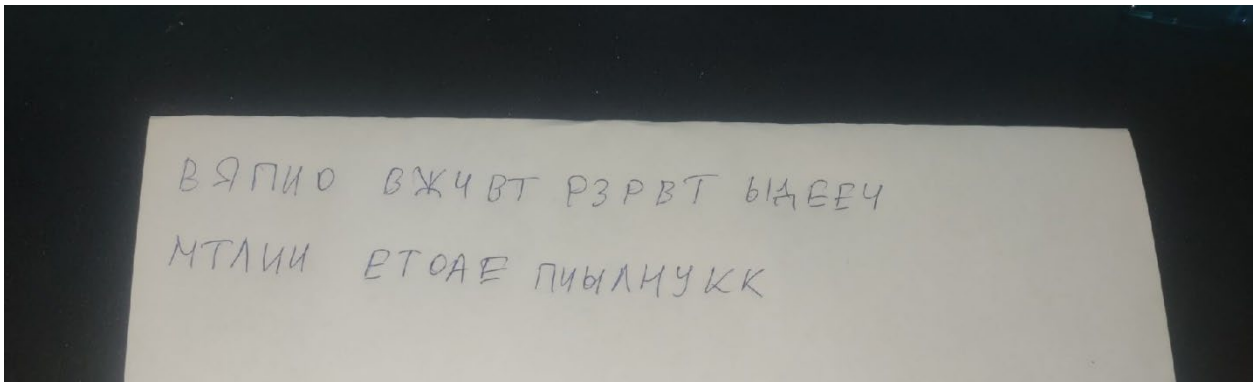
```
from base import alphabet, input_for_cipher_short, input_for_cipher_long, out
put_from_decrypted
import math

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:
врртаяилчотчаьттомндядачооннлгнляоифанпиикккеекквбехтацонидоокоинтаяилеевилвоив
юокичкаяилтоомовсикквзтсвчвяпдтдтоейиыынтиурряроисадиамчоивматлсснморачоарйтнин
саыиоалзупеуиабттинслесоиноомзялсомснокпеачнбпкэуеткаеофырисвиситозссосбмтто
двыетатнвриклтзиьтттеорспкеекоивжеаянвпетоисьтчмокданлкезпаохщлреввтеиааинлхо
цнукйвокроаодирбебонзлкнжбечтчмокнасьадлачданкстчмоолпесттсаааповякассяеививдв
чтоезпетегзомрмсиидлавзоитлееваекпйсдеопячтссбмитчрлвчеъеамотисилмолрыдеовдпт
смсарлкиеятктсмооккриижднпевттосаяилпеисаснаммочвооиясаептснебдгокттбтнлнуннтч
абрлчпеанссоттсомнйстиьодйкотририаиилбшнмохлцчаттдытеуихаичдогвчмозонссормооо
оьндлиуттенссоэккиноктиоытчыаютбоьяислрелнкплотлплмтзигчяеандилтоеовзнзтткрео
яьтиоттчрлиектбвлвомсрраививвнткзалсабыоавчиьбззиюзакпосчнntемивталмвсмьыувво
апипеенлнксноптчгиьчтийсзнпузиндчбштуцзсзозбв_тмтиыуввэотоеикптлпоияткаветмзх
деьириыбаттмсебелвлёзвыопаотонегкыувводнпзтиикануттыуввсьррлчатпзеттчлееттсл
есснеичзслобьдапюиуитрниисопкчвоиноститкелсртиьяпеибчеооеитекпенодтмоесьарем
вонрртчтоыаклттаотетдооырбиярдыаьиттчморлзчялижэемеегсятоссталыкеиопаноутоьс
нааыукеоок

Расшифрованный текст:
вотпримерстатьинатысячусимволов.этодостаточномаленькийтекст,оптимальноподходя
щийдлякарточектовароввинтернетилимагазинахилидлянебольшихинформационныхпублик
аций.втакомтекстередкобываетболеедвухилитрёхабзацевиобычноодинподзаголовок.но
можноибезнего.натысячусимволоврекомендованоиспользоватьодинилидваключаиоднука
ртину.текстнатысячусимволовэтосколькопримернослов.статистикапоказывает,чтотыс
ячавключаетвсебястопятьдесятилидвестисловсреднейвеличины.но,еслизлоупотреблят
ьпредлогами,союзамиидругимичастямиречинаодинилидвасимвола,токоличествословнеи
зменновозрастает.вкопирайтерскойдеятельностипринятосчитатьтысячиспробеламиили
без.учетпробеловувеличиваетобъемтекстапримернонастоилидвестисимволовименносто
лькоразмыразделяемсловасвободнымпространством.считатьпробелызаказчикинелюбят,
таккакэтопустоеместо.однаконекоторыефирмыибирживидятсправедливымставитьстоимо
стьзатысячусимволовспробелами,считаяпоследниеважнымэлементомкачественноговосп
риятия.согласитесь,читатьслитныйтекстбезединогопропуска,никтонебудет.нобольши
нствунужнаценазатысячузнаковбезпробелов.
```

Карточка:



11.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, out
put_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:
векаоврtетмячзплткпривиынелждутчивыелои
```

```
Расшифрованный текст:
время,приливыиотливынеждутчеловека.

ДЛИННЫЙ ТЕКСТ:
Зашифрованный текст:
чалекэтанвоькийоттгперкдсимыетосрсстзатятчоучпнсаимвоомлттьиовтнолимваряаопга
зионтщиамхвиалиьлвинойнитедплряндекоартотичлдхоектдиерейтчадянкобыквецбаоевтл
ьишбтаихокоомнинтыелкхнпублстиефоркацмдожнзагцоедибезонвеуехглоилвитовтриочкт
оёбчыккнчхноодоминабзнпоалвакьзоелатючаивоыксдянаучуосктьиммоадиевнндирлоован
оидитловспоровтчлькякинстатоиучтсчтпиктуекрикссмаеритмнвопсонловэлотоатыоскся
двестосскатислпозяыввсяраечтетьзпаддесвткялтниюччаетливетотсебыясоютьптзйвами
ирдеелриуегчиснидлытломгаичзмлиизокупотптрчнозеблптнеиокоизасменнлотлявмоизир
иерчечидсатввнаоссслиамволоватодизптниысяприрчаеиспрнотстбчеялквккатоопосмчий
идтеаитяртелььтниайтостебнонъемеалистоитлблеиздевтчокексучвтсапуевреитмлтичив
ераипроетобямпремсоосистралнмлвсотовлоьвоваимксмвооербаотдзнмыранызчносделтпе
меттаасксстотчккчкиотдкнатаьаакпрзэкточоипкуосибнелютобнелыятззиоствымрьекзаты
ссожтяочтурыиесавфивиитьирдсятмотмспраиммвыибедливачеажнтсолтвеныносвосгмопрч
овэлбеиеометланятсопаослемкдпмизниептогоныйтприропутсяеткиаезятсчпкскскьттбезо
пзтендчгитатиньиласслииевбеназоктпробаеолнлеотвбуьдтысетшяччуитнзснкатчвунук
ож кнонацб

Расшифрованный текст:
вотпримерстатьинатысячусимволов.этодостаточномаленькийтекст,оптимальноподходя
щийдлякарточектовароввинтернетилимагазинахилидлянебольшихинформационныхпублик
аций.втакомтекстередкобываетболеедвухилитрёхабзацевиобычноодинподзаголовок.но
можноибезнего.натысячусимволоврекомендованоиспользоватьодинилидваключаиоднука
ртину.текстнатысячусимволовэтосколькопримернослов.статистикапоказывает,чтотыс
ячавключаетвсебястопятьдесятилидвестисловсреднейвеличины.но,еслизлоупотреблят
ьпредлогами,союзамиидругимичастямиречинаодинилидвасимвола,токоличествословнеи
зменновозрастает.вкопирайтерскойдеятельностипринятосчитатьтысячиспробеламиили
без.учетпробеловувеличиваетобъемтекстапримернонастоилидвестисимволовименностоо
лькоразмыразделяемсловасвободнымпространством.считатьпробелызаказчикинелюбят,
таккакэтопустоеместо.однаконекоторыефирмыибиржевидятсправедливымставитьстоимо
стьзатысячусимволовспробелами,считаяпоследниеважнымэлементомкачественноговосп
риятия.согласитесь,читатьслитныйтекстбезединогопропуска,никтонебудет.нобольши
нствунужнаценазатысячузнаковбезпробелов.
```

Карточка: