

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
Постоянный модуль	4
Блок А: ШИФРЫ ОДНОЗНАЧНОЙ ЗАМЕНЫ.....	5
1. Шифр простой замены АТБАШ.....	5
2. ШИФР ЦЕЗАРЯ.....	7
3. Квадрат Полибия	10

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 ', '!': 'вскл ', '?': 'вп  
р ' }

def replace_all_to(input_text, dict):
    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)
```

Блок А: ШИФРЫ ОДНОЗНАЧНОЙ ЗАМЕНЫ

1. Шифр простой замены АТБАШ

Атбаш — простой шифр подстановки для алфавитного письма. Правило шифрования состоит в замене i -й буквы алфавита буквой с номером $n-i+1$, где n — число букв в алфавите.

Код программы:

```
from base import alphabet, input_for_cipher_short, input_for_cipher_long, output_from_decrypted

# функция преобразования текста
def atbash(input):
    return input.translate(str.maketrans(
        alphabet + alphabet.upper(), alphabet[::-1] + alphabet.upper()[::-1]))

# вывод результатов работы программы
print(f'''
ШИФР АТБАШ:
КОРОТКИЙ ТЕКСТ:
Зашифрованный текст:
{atbash(input_for_cipher_short())}

Расшифрованный текст:
{output_from_decrypted(atbash(atbash(input_for_cipher_short())))}
'''

ДЛИННЫЙ ТЕКСТ:
Зашифрованный текст:
{atbash(input_for_cipher_long())}

Расшифрованный текст:
{output_from_decrypted(atbash(atbash(input_for_cipher_long())))}
''')
```

Тестирование:

```
root@DESKTOP-05UI9FD:~/crypt# /bin/python3
/root/crypt/lab01_1_atbash.py
ШИФР АТБАШ:
КОРОТКИЙ ТЕКСТ:
Зашифрованный текст:
эоъта чпм поцуцэд ц рмуцэд съ шылм зъурэъфя мзф

Расшифрованный текст:
время, приливы и отливы не ждут человека.
```

ДЛИННЫЙ ТЕКСТ:

Зашифрованный текст:

эрм поцтѣо нмямгц ся мдназл нцтѣрурэ мзф вmr ьрнмямрзср
тяуѣстгфцх мѣфнм чпм рпмцтяугср прѣйрыаѣцх ьуа фяомрзѣф мрѣяорѣ э
цсмѣосѣм цуц тѣячцсѣя цуц ьуа сѣюругжцѣ цскротѣяцрссдѣ
плѣуцфѣяцх мзф э мѣфрт мѣфнмѣ оѣфр юдѣяѣм юруѣѣ ьѣлѣ цуц моѣѣ
яючѣяѣѣ ц рюдзср рыцс прѣчѣярурѣрѣф мзф ср тршср ц юѣч сѣър мзф
сѣ мдназл нцтѣрурэ оѣфртѣсѣрѣяср цнпругчрѣямг рыцс цуц ьѣя фубзѣ
ц рысл фяомцсл мзф мѣфнм сѣ мдназл нцтѣрурэ вmr нфругфр поцтѣоср
нурѣ эпо нмямцнмцѣя прѣячдѣяѣм чпм зmr мдназѣ эфубзѣяѣм э нѣюа
нмрпамгѣѣнам цуц ьѣѣнмц нурѣ ноѣысѣх эѣуцццсд мзф ср чпм ѣнуц
чурлпрмоѣюуамг поѣуурѣятц чпм нрбчѣятц ц ьольцтц зѣяматц оѣзц сѣ
рыцс цуц ьѣя нцтѣруѣя чпм мр фруцзѣнмѣр нурѣ сѣцчтѣсср эрчоянмяѣм
мзф э фрпцояхмѣонфрх ьѣамѣугсрнмц поцсамр нзцмямг мдназц н
порѣюуѣятц цуц юѣч мзф лзѣм порѣюурѣ лѣѣуццзѣяѣм рѣѣѣт мѣфнмѣ
поцтѣоср сѣ нmr цуц ьѣѣнмц нцтѣрурэ цтѣсср нмругфр оѣч тд
оѣчѣѣуѣѣт нурѣя нѣрюысдт порнмояснмѣрт мзф нзцмямг порѣюуд
чѣфѣчзцфц сѣ убѣюам чпм мѣф фѣф вmr плнмрѣ тѣнмр мзф рысѣфр
сѣфрмродѣ кцотд ц юцощ эцѣям нпояѣѣуцѣдт нмяѣцмг нмрцтрнмг чѣ
мдназл нцтѣрурэ н порѣюуѣятц чпм нзцмѣя прнѣуѣысѣѣ эѣшсдт
вѣѣтѣсмрт фѣзѣнмѣѣссрѣр эрнпоѣамца мзф нрѣуѣянцмѣнг чпм зцмямг
нуцмсдх мѣфнм юѣч ѣѣцсрѣр порплнѣя чпм сѣфмр сѣ юѣѣм мзф ср
юругжцснмѣл слшсѣя иѣсѣя чѣ мдназл чсѣфрѣ юѣч порѣюурѣ мзф

Расшифрованный текст:

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 предложениями, 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.

Исполняемый файл: https://github.com/ISenichl/mospolytech-education-crypt-dev-2021-1/blob/main/lab01_1_atbash.py

2. ШИФР ЦЕЗАРЯ

Шифр Цезаря, также известный как шифр сдвига, код Цезаря или сдвиг Цезаря — один из самых простых и наиболее широко известных методов шифрования.

Шифр Цезаря — это вид шифра подстановки, в котором каждый символ в открытом тексте заменяется символом, находящимся на некотором постоянном числе позиций левее или правее него в алфавите. Например, в шифре со сдвигом 3 А была бы заменена на Г, Б станет Д, и так далее.

Используемое преобразование обычно обозначают как ROTN, где N — сдвиг, ROT — сокращение от слова ROTATE, в данном случае «циклический сдвиг». Число разных преобразований конечно и зависит от длины алфавита.

Код программы:

```
from base import alphabet, input_for_cipher_short, input_for_cipher_long, output_from_decrypted
import random

# случайное число для сдвига
rand_int = random.randint(1, 32)

# функция шифровки
def caesar_encode(input, step):
    return input.translate(
        str.maketrans(alphabet, alphabet[step:] + alphabet[:step]))

# функция дешифровки
def caesar_decode(input, step):
    return input.translate(
        str.maketrans(alphabet[step:] + alphabet[:step], alphabet))

# вывод результатов работы программы
print(f'''
ШИФР ЦЕЗАРЯ:
КОРОТКИЙ ТЕКСТ:
Зашифрованный текст:
{caesar_encode(input_for_cipher_short(), rand_int)}

Расшифрованный текст:
{output_from_decrypted(caesar_decode(caesar_encode(
    input_for_cipher_short(), rand_int), rand_int))}

ДЛИННЫЙ ТЕКСТ:
Зашифрованный текст:
```

```
{caesar_encode(input_for_cipher_long(), rand_int)}

Расшифрованный текст:
{output_from_decrypted(caesar_decode(caesar_encode(
    input_for_cipher_long(), rand_int), rand_int))}
'''}
```

Тестирование:

```
root@DESKTOP-05UI9FD:~/crypt# /bin/python3
/root/crypt/lab01_2_caesar.py
```

ШИФР ЦЕЗАРЯ:

КОРОТКИЙ ТЕКСТ:

Зашифрованный текст:

бпдлю жос опзкзбъ з нскзбъ мд ёгтс цдкнбдйя сцй

Расшифрованный текст:

время, приливы и отливы не ждут человека.

ДЛИННЫЙ ТЕКСТ:

Зашифрованный текст:

бнс опзлдп рсясыз мя сърюцт рзлбнкнб сцй ьсн гнрсяснцмн
лякдмьйзи сдйрс жос носзлякымн онгфнгюшзи гкю йяпснцдй снбяпнб б
змсдпмдс экз лявяжзмяф экз гкю мданкычзф змунпляхзнммъф
отакзйяхзи сцй б саянл сдйрсд пдгйн аъбядс анкдд гбтф экз спеф
яажяхдб з наъцмн нгзм онгжявнкнбнй сцй мн лнёмн з адж мдвн сцй
мя сърюцт рзлбнкнб пдйнлмгнбямн зронкыжнбясы нгзм экз гбя йкэця
з нгмт йяпсзт сцй сдйрс мя сърюцт рзлбнкнб ьсн рйнкыйн опзлдпмн
ркнб боп рсясзрсзйя онйяжъбядс жос цсн сърюця бйкэцядс б рдаю
рсн оюсыгдрюс экз гбдрсз ркнб рпдгмди бдкзцзмъ сцй мн жос дркз
жкнтонспдакюсы опдгкнвялз жос рнэжялз з гптвзлз цярсюлз пдцз мя
нгзм экз гбя рзлбнкя жос сн йнкзцдрсбн ркнб мдзжлдмн бнжпярсядс
сцй б йнозпяисдприни гдюдскымнрсз опзмюсн рцзсясы сърюцз р
опнадкялз экз адж сцй тцдс опнадкнб тбдкзцзбядс нащдл сдйрся
опзлдпмн мя рсн экз гбдрсз рзлбнкнб злдмн рснкийн пяз лъ
пязгдкюдл ркнбя рбнангмъл опнрспямрсбнл сцй рцзсясы опнадкъ
жййяжцзйз мд кэаюс жос сйй ййй ьсн отрснд лдрсн сцй нгмйин
мдйнснпъд узплъ з азпёз бзгюс ропябдгкзбъл рсябзсы рснзлнрсы жя
сърюцт рзлбнкнб р опнадкялз жос рцзсяю онркдгмзд бяёмъл
ькдлдмснл йяцдрсбдмнвн бнропзюсзю сцй рнвкярзсдры жос цзсясы
ркзсмьи сдйрс адж дгзмнвн опнотрйя жос мзйсн мд атгдс сцй мн
анкычзмрсбт мтёмя хдмя жя сърюцт жмяйнб адж опнадкнб сцй

Расшифрованный текст:

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.

Исполняемый файл: https://github.com/lSenichl/mospolytech-education-crypt-dev-2021-1/blob/main/lab01_2_caesar.py

3. Квадрат Полибия

Квадрат Полибия – метод шифрования текстовых данных с помощью замены символов, впервые предложен греческим историком и полководцем Полибием.

К каждому языку отдельно составляется таблица шифрования с одинаковым (не обязательно) количеством пронумерованных строк и столбцов, параметры которой зависят от его мощности (количества букв в алфавите). Берутся два целых числа, произведение которых ближе всего к количеству букв в языке — получаем нужное число строк и столбцов. Затем вписываем в таблицу все буквы алфавита подряд — по одной на каждую клетку. При нехватке клеток можно вписать в одну две буквы (редко употребляющиеся или схожие по употреблению).

Код программы:

```
from base import alphabet, input_for_cipher_short, input_for_cipher_long, output_from_decrypted

# квадрат полибия
hard_dictionary = {"a": "11", "б": "12", "в": "13",
                  "г": "14", "д": "15", "е": "16", "ё": "21",
                  "ж": "22", "з": "23", "и": "24", "й": "25",
                  "к": "26", "л": "31", "м": "32", "н": "33",
                  "о": "34", "п": "35", "р": "36", "с": "41",
                  "т": "42", "у": "43", "ф": "44", "х": "45",
                  "ц": "46", "ч": "51", "ш": "52", "щ": "53",
                  "ъ": "54", "ы": "55", "ь": "56", "э": "61",
                  "ю": "62", "я": "63"}

# функция шифровки
def square_encode(input):
    new_txt = ""
    for x in input:
        if x in hard_dictionary:
            new_txt += hard_dictionary.get(x)
        else:
            new_txt += (x + x)
    return new_txt

# функция дешифровки
def square_decode(input):
    new_txt = ""
    list_fraze = []
    step = 2
    for i in range(0, len(input), 2):
        list_fraze.append(input[i:step])
        step += 2
```

```

key_hard_dictionary_list = list(hard_dictionary.keys())
val_hard_dictionary_list = list(hard_dictionary.values())

for x in list_fraze:
    if x in val_hard_dictionary_list:
        i = val_hard_dictionary_list.index(x)
        new_txt += key_hard_dictionary_list[i]
    else:
        new_txt += x[0:1]
return new_txt

# вывод результатов работы программы
print(f'''
КВАДРАТ ПОЛИБИЯ:
КОРОТКИЙ ТЕКСТ:
Зашифрованный текст:
{square_encode(input_for_cipher_short())}

Расшифрованный текст:
{output_from_decrypted(square_decode(square_encode(
    input_for_cipher_short())))}

ДЛИННЫЙ ТЕКСТ:
Зашифрованный текст:
{square_encode(input_for_cipher_long())}

Расшифрованный текст:
{output_from_decrypted(square_decode(square_encode(
    input_for_cipher_long())))}
''')

```

Тестирование:

```

root@DESKTOP-05UI9FD:~/crypt# /bin/python3
/root/crypt/lab01_3_square.py

КВАДРАТ ПОЛИБИЯ:
КОРОТКИЙ ТЕКСТ:
Зашифрованный текст:
1336163263  233542  35362431241355  24  344231241355  3316
22154342  5116313413162611  425126

Расшифрованный текст:
время, приливы и отливы не ждут человека.

ДЛИННЫЙ ТЕКСТ:
Зашифрованный текст:
133442  353624321636  414211425624  3311  425541635143
4124321334313413  425126  614234  15344142114234513334

```

321131163356262425 4216264142 233542 34354224321131563334
35341545341563532425 153163 2611364234511626 42341311363413
13 2433421636331642 243124 321114112324331145 243124 153163
331612343156522445 2433443436321146243433335545
35431231242611462425 425126 13 4211263432 421626414216
3616152634 125513111642 1234311616 15134345 243124 42362145
11122311461613 24 341255513334 34152433
353415231114343134133426 425126 3334 3234223334 24 121623
33161434 425126 3311 425541635143 4124321334313413
36162634321633153413113334 244135343156233413114256 34152433
243124 151311 2631625111 24 34153343 26113642243343 425126
4216264142 3311 425541635143 4124321334313413 614234
41263431562634 3536243216363334 41313413 133536
41421142244142242611 35342611235513111642 233542 514234
425541635111 1326316251111642 13 41161263 414234
356342561516416342 243124 151316414224 41313413
41361615331625 1316312451243355 425126 3334 233542 16413124
2331344335344236161231634256 35361615313414113224 233542
41346223113224 24 15364314243224 51114142633224 36165124
3311 34152433 243124 151311 41243213343111 233542 4234
26343124511641421334 41313413 331624233216333334
13342336114142111642 425126 13 2634352436112542163641263425
151663421631563334414224 35362433634234 41512442114256
425541635124 41 353634121631113224 243124 121623 425126
43511642 3536341216313413 4313163124512413111642 3412541632
421626414211 3536243216363334 3311 414234 243124
151316414224 4124321334313413 243216333334 41423431562634
361123 3255 361123151631631632 4131341311 411334123415335532
35363441423611334142133432 425126 41512442114256
35363412163155 231126112351242624 3316 3162126342 233542
421126 261126 614234 354341423416 3216414234 425126
341533112634 331626344234365516 4424363255 24 1224362224
1324156342 413536111316153124135532 41421113244256
414234243234414256 2311 425541635143 4124321334313413 41
353634121631113224 233542 415124421163 353441311615332416
131122335532 613116321633423432 26115116414213163333341434
13344135362463422463 425126 4134143111412442164156 233542
512442114256 41312442335525 4216264142 121623 16152433341434
3536343543412611 233542 3324264234 3316 1243151642 425126
3334 1234315652243341421343 3343223311 46163311 2311
425541635143 233311263413 121623 3536341216313413 425126

Расшифрованный текст:

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.

Исполняемый файл: https://github.com/lSenichl/mospolytech-education-crypt-dev-2021-1/blob/main/lab01_3_square.py