Аннотация

Среда программирования: Visual Studio Code

Язык программирования: Python 3

Процедуры для запуска программы: $ python3 <имя_файла>.py

Пословица-тест: Красивыми словами пастернак не помаслишь

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#

Блок Е: ШИФРЫ ГАММИРОВАНИЯ

- **Гаммирование ГОСТ 28147-89**

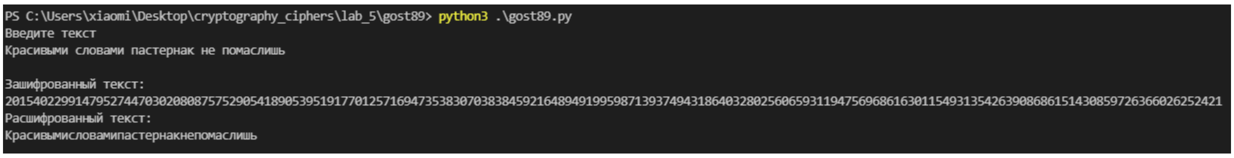
Код программы:

```
# -*- coding:utf-8 -*-
import sys
import numpy.random
import itertools
from demo import alphabet, input_for_cipher_short, input_for_cipher_long,
output_from_decrypted
import binascii


class GostCrypt(object):
    def __init__(self, key, sbox):
        self._key = None
        self._subkeys = None
        self.key = key
        self.sbox = sbox

    @staticmethod
    def _bit_length(value):
        return len(bin(value)[2:])

    @property
    def key(self):
        return self._key

    @key.setter
    def key(self, key):
        self._key = key
        self._subkeys = [(key >> (32 * i)) & 0xFFFFFFFF for i in range(8)] #8
кусков


    def _f(self, part, key):
        temp = part ^ key
        output = 0
        for i in range(8):
            output |= ((self.sbox[i][(temp >> (4 * i)) & 0b1111]) << (4 * i))
        return ((output >> 11) | (output << (32 - 11))) & 0xFFFFFFFF


    def _decrypt_round(self, left_part, right_part, round_key):
        return left_part, right_part ^ self._f(left_part, round_key)

    def encrypt(self, plain_msg):
        def _encrypt_round(left_part, right_part, round_key):
            return right_part, left_part ^ self._f(right_part, round_key)
```

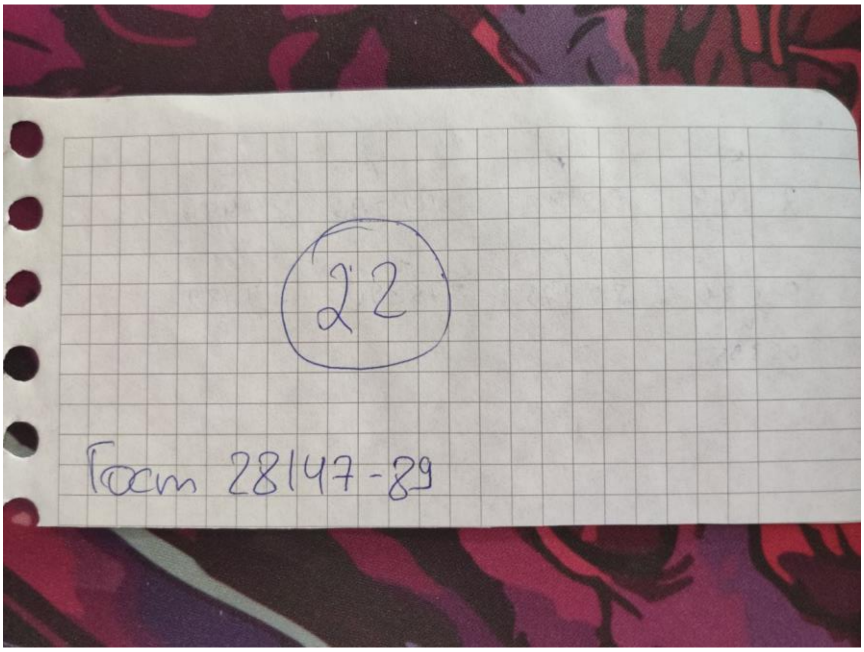
```
        left_part = plain_msg >> 32
        right_part = plain_msg & 0xFFFFFFFF
        for i in range(24):
            left_part, right_part = _encrypt_round(left_part, right_part,
self._subkeys[i % 8])
        for i in range(8):
            left_part, right_part = _encrypt_round(left_part, right_part,
self._subkeys[7 - i])
        return (left_part << 32) | right_part

    def decrypt(self, crypted_msg):
        def _decrypt_round(left_part, right_part, round_key):
            return right_part ^ self._f(left_part, round_key), left_part

        left_part = crypted_msg >> 32
        right_part = crypted_msg & 0xFFFFFFFF
        for i in range(8):
            left_part, right_part = _decrypt_round(left_part, right_part,
self._subkeys[i])
        for i in range(24):
            left_part, right_part = _decrypt_round(left_part, right_part,
self._subkeys[(7 - i) % 8])
        return (left_part << 32) | right_part


sbox = [numpy.random.permutation(l) for l in itertools.repeat(list(range(16)),
8)]
sbox = (
    (4, 10, 9, 2, 13, 8, 0, 14, 6, 11, 1, 12, 7, 15, 5, 3),
    (14, 11, 4, 12, 6, 13, 15, 10, 2, 3, 8, 1, 0, 7, 5, 9),
    (5, 8, 1, 13, 10, 3, 4, 2, 14, 15, 12, 7, 6, 0, 9, 11),
    (7, 13, 10, 1, 0, 8, 9, 15, 14, 4, 6, 12, 11, 2, 5, 3),
    (6, 12, 7, 1, 5, 15, 13, 8, 4, 10, 9, 14, 0, 3, 11, 2),
    (4, 11, 10, 0, 7, 2, 1, 13, 3, 6, 8, 5, 9, 12, 15, 14),
    (13, 11, 4, 1, 3, 15, 5, 9, 0, 10, 14, 7, 6, 8, 2, 12),
    (1, 15, 13, 0, 5, 7, 10, 4, 9, 2, 3, 14, 6, 11, 8, 12),
    )

key =
18318279387912387912789378912379821879387978238793278872378329832982398023031

text_short = input_for_cipher_short().encode().hex()
text_short = int(text_short, 16)

gost_short = GostCrypt(key, sbox)

enc_txt = gost_short.encrypt(text_short)
dec_txt = gost_short.decrypt(enc_txt)
dec_txt = bytes.fromhex(hex(dec_txt)[2::]).decode('utf-8')
```

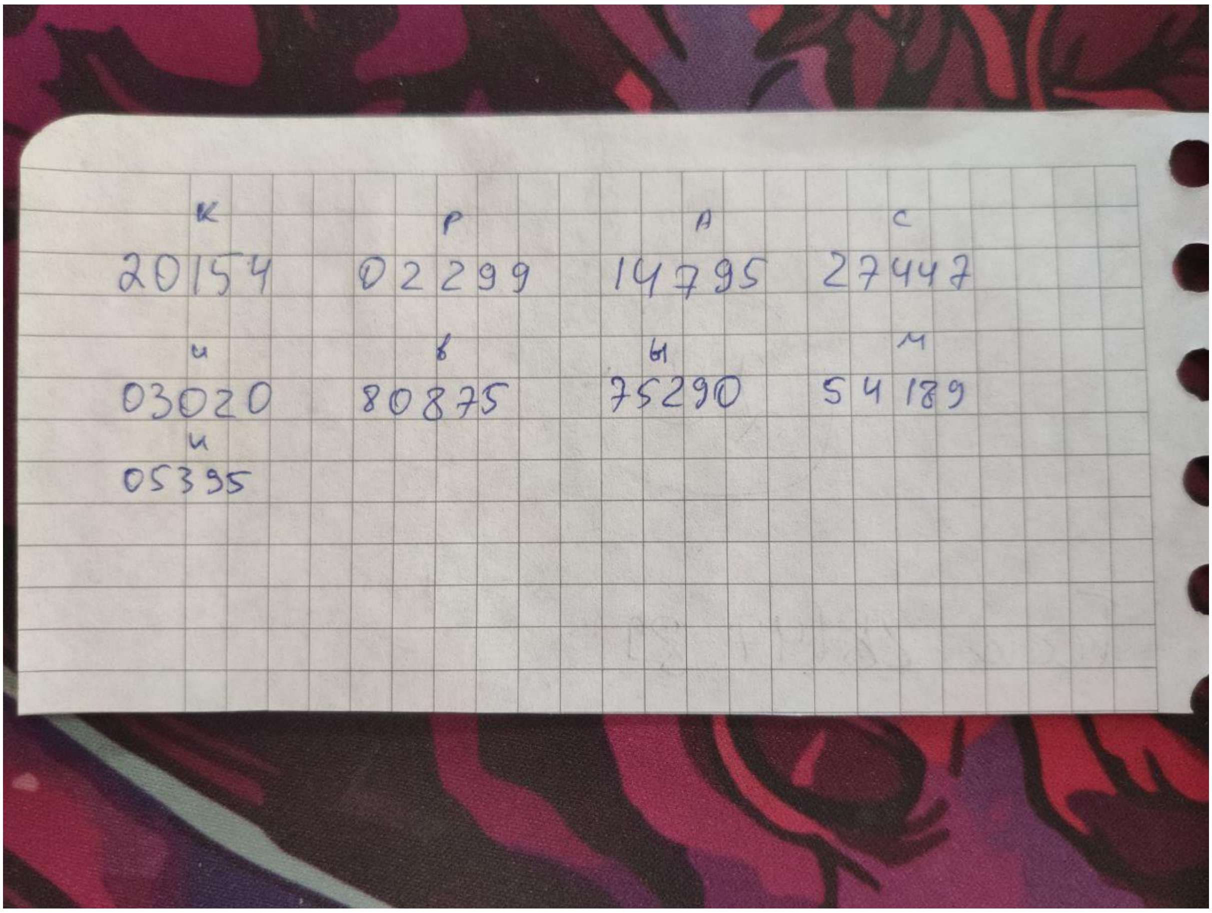
```python
text_long = input_for_cipher_long().encode().hex()
text_long = int(text_long, 16)

print(f'''
Зашифрованный текст:
{enc_txt}
Расшифрованный текст:
{output_from_decrypted(dec_txt)}
''')
```

Тестирование:

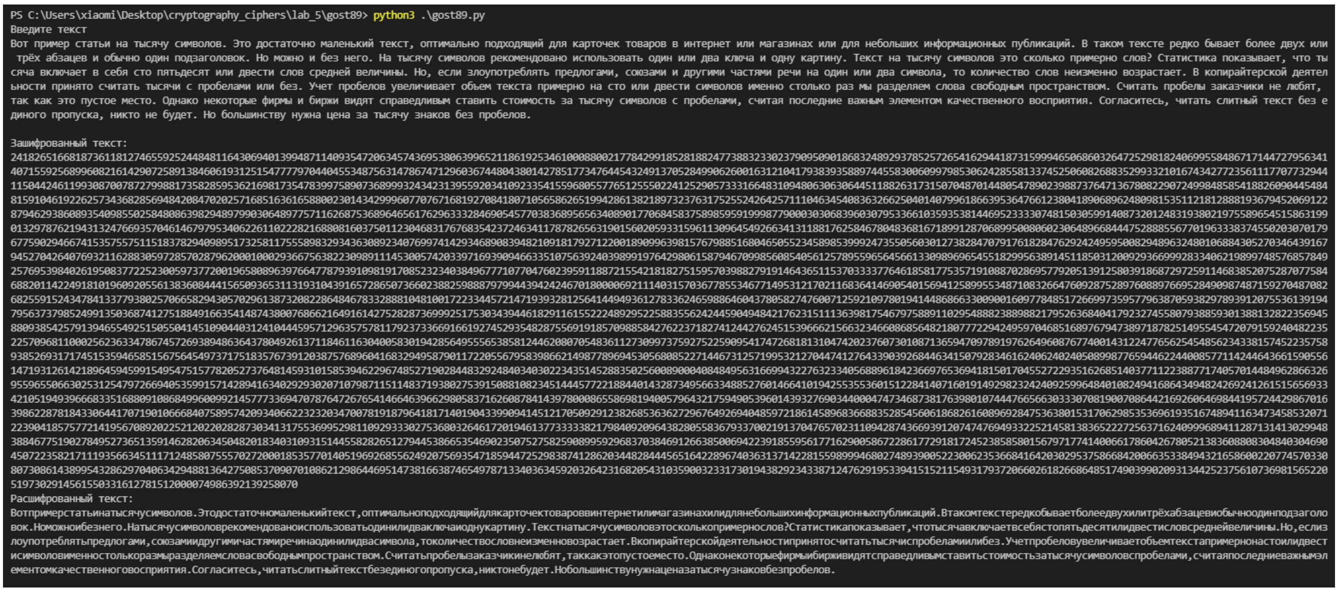


```
PS C:\Users\xiaomi\Desktop\cryptography_ciphers\lab_5\gost89> python3 .\gost89.py
Введите текст
Красивыми словами пастернак не помаслишь

Зашифрованный текст:
201540229914795274470302080875752905418905395191770125716947353830703838459216489491995987139374943186403280256065931194756968616301154931354263908686151430859726366026252421
Расшифрованный текст:
Красивымисловамипастернакнепомаслишь
```





к р а с
20154 02299 14795 27447
и в ы м
03020 80875 75290 54189
и
05395



```
PS C:\Users\xiaomi\Desktop\cryptography_ciphers\lab_5\gost89> python3 .\gost89.py
Введите 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 ты
сяча 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 деятел
ьности 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 е
диного пропуска, никто не будет. Но большинству нужна цена за тысячу знаков без пробелов.

Зашифрованный текст:
241826516681873611812746559252448481164306940139948711409354720634574369538063996521186192534610008800217784299185281882477388323302379095090186832489293785257265416294418731599946506860326472529818240699558486717144727956341
407155925689960821614290725891384606193125154777797044045534875631478674712960367448043801427851773476445432491370528499062600163121041793839358897445583006099798530624285581337452506082688352993321016743427723561117707732944
115044246119930870078727998817358285953621698173547839975890736899932434231395592034109233541559680557765125550224125290573331664831094806306306445118826317315070487014480547890239887376471367808229072499848585418826090445484
815910461922625734368285694842084702025716851636165880023014342999607707671681927084180710565862651994286138218973237631752552426425711104634540836326625040140799618663953647661238041890689624809815351121812888193679452069122
879462938608935409855025848086398294897990306489775711626875368964656176296333284690545770383689565634089017706845837589859591999877900030306839603079533661035935381446952333307481503059914087320124831938021975589654515863199
013297876219431324766935704614679795340622611022282168808160375011230468317676835423724634117878265631901560205933159611309645492663413118817625846780483681671899128706899500806023064896684447528885567701963338374550203070179
677590294667415357557511518378294098951732581175558983293436308923407699741429346890839482109181792712200189099639815767988516804650552345898539992473550560301273828470791761828476292424959500829489632480106884305270346439167
945270426407693211628830597285702879620001000293667563822309891114530057420339716939094663351075639240398991976429806158794670998560854056125789559656456613309896965455182995638914511850312009293669992833406219899748576857849
257695398402619508377225230059737720019658089639766477879391098191708523234038496777107704760239591188721554218182751595703988279191464365115370333377646185817753571910887028695779205139125803918687297259114683852075287077584
688201142249181019609205561383608444156509365311319310439165728650736602388259888797994439424246701800006921114031570367785534677149531217021168364146905401569412589955348710832664760928752897608897669528490987487159270487082
682559152434784133779380257066582943057029613873208228648467833288810481001722334457214719393281256414494936127833624659886460437805827476007125921097801941448686633009001609778485172669973595779638705938297893912075536139194
795637379852499135036874127518849166354148743800768662164916142752828736999251753034394461829116155222489295225883556242445904948421762315111363981754679758891102954888238898821795263684041792327455807938859301388132822356945
880938542579139465549251505504145109044031241044459571296357578117923733669166192745293548287556919185709885842762237182741244276245153966621566323466086856482180777229424959704685168976794738971878251495545472079159240482235
225709681100025623633478674572693894863643780492613711846116304005830194285649555653858124462080705483611273099737592752259095417472681813104742023760730108713659470978919762649608767740014312247765625454856234338157452235758
938526931717451535946585156756454973717518357673912038757689604168329495879011722055679583986621498778969453056808522714467312571995321270447412764339039268446341507928346162406240240508998776594462244008577114244643661590556
147193126142189645945991549547515778205273764814593101585394622967485271902844832924840340302234351452883502560089000408484956316699432276323340568896184236697653694181501704552722935162685140377112238877174057014484962866326
955965506630253125479726694053599157142894163402929302071079871151114837193802753915088108234514445772218844014328734956633488527601466410194255355360151228414071601914929823242409259964840108249416864349482426924126151565693 3
421051949396668335168809108684996009921457773369470787647267654146646396629805837162608784143978000865586981940057964321759490539601439327690344000474734687381763980107444766566303330708190070864421692606469844195724429867016
398622878184330644170719010666840758957420934066223232034700781918796418171401904339909414512170509291238268536362729676492694048597218614589683668835285456061868261608969284753638015317062985353696193516748941163473458532071
223904185757721419567089202252120220282873034131755369952981109293330275368032646172019461377333338217984092096438280558367933700219137047657023110942874366939120747476949332252145813836522272563716240999689411287131413029948
388467751902784952736513591462820634504820183403109315144558282651279445386653546902350752758259089959296837038469126638500694223918555956177162900586722861772918172452385858015679717741400661786042678052138360880830484030469 0
450722358217111935663451117124858075557027200018535770140519692685562492075693547185944725298387412862034482844456516422896740363137142281559899946802748939005223006235366841642030295375866842006635338494321658600220774570330
807308614389954328629704063429488136427508537090701086212986446951473816638746549787133403634592032642316820543103590032331730194382923433871247629195339415152115493179372066026182668648517490399020931344252375610736981565220
5197302914561550331612781512000074986392139258070
Расшифрованный текст:
Вотпримерстатьинатысячусимволов.Этодостаточномаленькийтекст,оптимальноподходящийдлякарточектовароввинтернетилимагазинахилидлянебольшихинформационныхпубликаций.Втакомтекстередкобываетболеедвухилитрёхабзацевиобычноодинподзаголо
вок.Номожноибезнего.Натысячусимволоврекомендованоиспользоватьодинилидваключаиоднукартину.Текстнатысячусимволовэтосколькопримернослов?Статистикапоказывает,чтотысячавключаетвсебястопятьдесятилидвестисловсреднейвеличины.Но,еслиз
лоупотреблятьпредлогами,союзамиидругимичастямиречинаодинилидвасимвола,токоличествословнеизменновозрастает.Вкопирайтерскойдеятельностипринятосчитатьтысячиспробеламиилибез.Учетпробеловувеличиваетобъемтекстапримернонастоилидвест
исимволовименностолькоразмыразделяемсловасвободнымпространством.Считатьпробелызаказчикинелюбят,таккакэтопустоеместо.Однаконекоторыефирмыибирживидятсправедливымставитьстоимостьзатысячусимволовспробелами,считаяпоследниеважнымэл
ементомкачественноговосприятия.Согласитесь,читатьслитныйтекстбезединогопропуска,никтонебудет.Нобольшинствунужнаценазатысячузнаковбезпробелов.
```

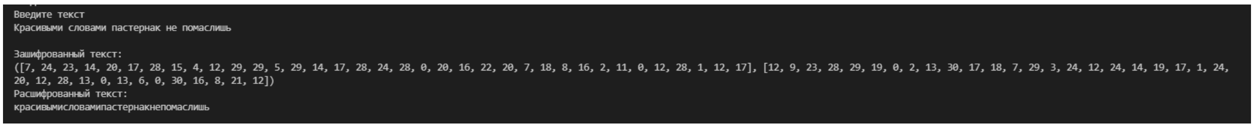
- **Одноразовый блокнот К.Шеннона**

Код программы:

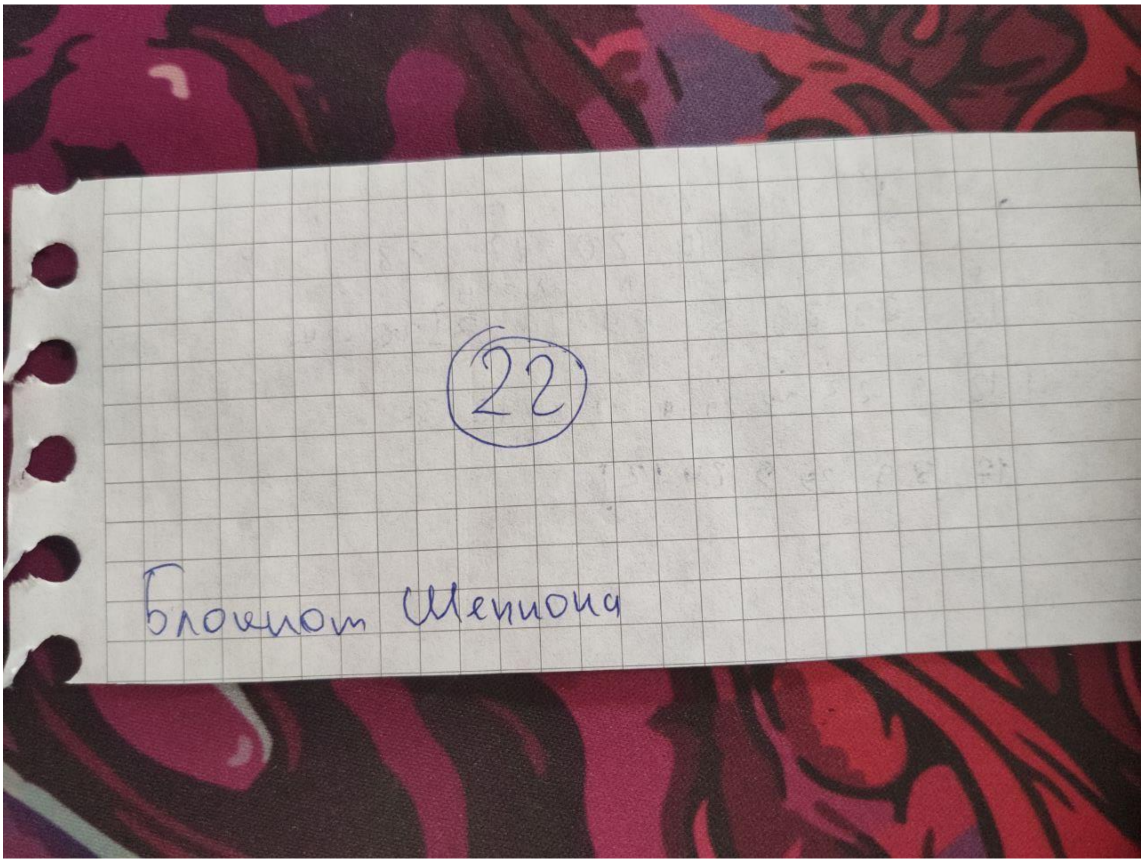
```
import random

alphabet = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя "

alphabet = alphabet.replace(' ', '')
alphabet_lower = {}
```

```
i = 0
while i < (len(alphabet)):
    alphabet_lower.update({alphabet[i]: i})
    i += 1


def get_key(d, value):
    for k, v in d.items():
        if v == value:
            return k


def encode(msg):
    msg_list = list(msg)
    msg_list_len = len(msg_list)
    msg_code_bin_list = list()
    for i in range(len(msg_list)):
        msg_code_bin_list.append(alphabet_lower.get(msg_list[i]))

    key_list = list()
    for i in range(msg_list_len):
        key_list.append(random.randint(0, 32))

    cipher_list = list()
    for i in range(msg_list_len):
        m = int(msg_code_bin_list[i])
        k = int(key_list[i])
        cipher_list.append(int(bin(m ^ k), base=2))
    return cipher_list, key_list


def decode(msg, key_list):
    decipher_list = list()
    msg_list_len = len(msg)
    for i in range(msg_list_len):
        c = int(msg[i])
        k = int(key_list[i])
        decipher_list.append(int(bin(c ^ k), base=2))
    deciphered_str = ""
    for i in range(len(decipher_list)):
        deciphered_str += get_key(alphabet_lower, decipher_list[i])
    return deciphered_str
```

Тестирование:

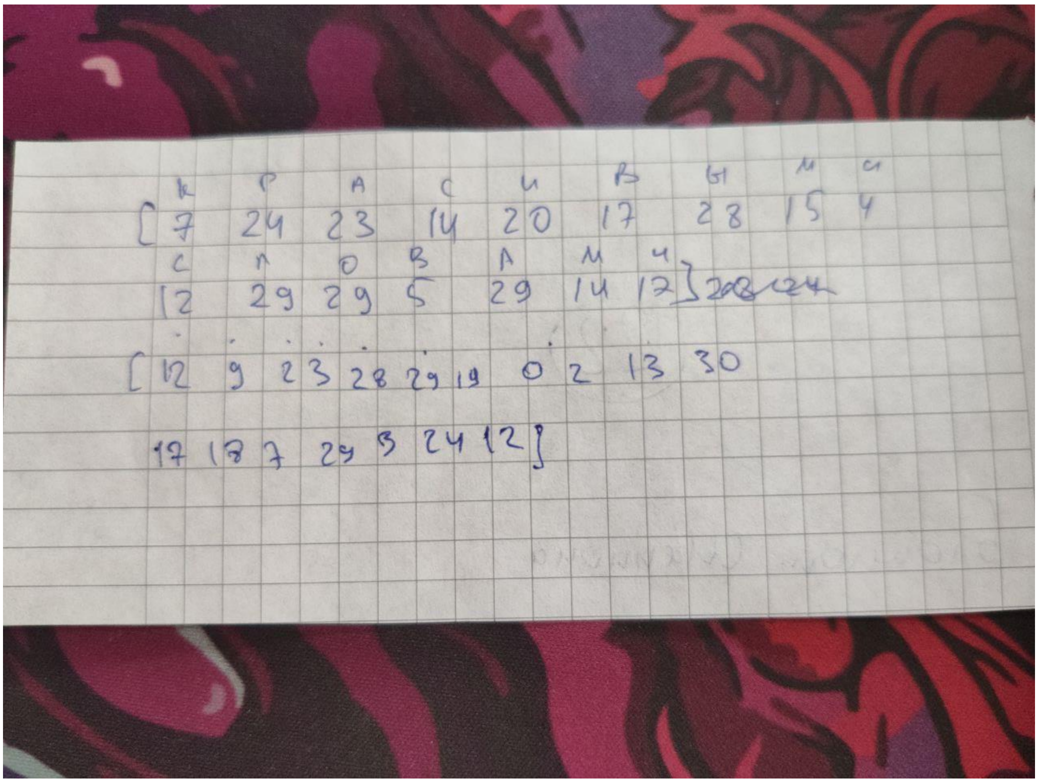


```
Введите текст
Красивыми словами пастернак не помаслишь

Зашифрованный текст:
([7, 24, 23, 14, 20, 17, 28, 15, 4, 12, 29, 29, 5, 29, 14, 17, 28, 24, 28, 0, 20, 16, 22, 20, 7, 18, 8, 16, 2, 11, 0, 12, 28, 1, 12, 17], [12, 9, 23, 28, 29, 19, 0, 2, 13, 30, 17, 18, 7, 29, 3, 24, 12, 24, 14, 19, 17, 1, 24,
20, 12, 28, 13, 0, 13, 6, 0, 30, 16, 8, 21, 12])
Расшифрованный текст:
красивымисловамипастернакнепомаслишь
```



22

Блокнот Шпиона



к р а с и в ы м и
[7 24 23 14 20 17 28 15 4
с л о в а м и
12 29 29 5 29 14 12] [illegible]

[12 9 23 28 29 19 0 2 13 30

17 18 7 29 13 24 12]

```
Введите 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 ты
сяча 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 деятел
ьности 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,


Зашифрованный текст:
([18, 2, 17, 8, 21, 25, 0, 13, 12, 5, 19, 20, 31, 10, 31, 28, 27, 24, 6, 31, 54, 22, 52, 22, 5, 1, 1, 9, 7, 23, 8, 2, 2, 7, 3, 25, 22, 13, 5, 26, 16, 2, 9, 15, 23, 31, 14, 3, 20, 23, 22, 16, 7, 1, 8, 6, 24, 20, 43, 29, 16, 7,
 31, 20, 8, 20, 23, 2, 7, 7, 31, 7, 23, 11, 16, 22, 6, 26, 1, 2, 49, 20, 21, 19, 8, 20, 60, 5, 9, 0, 23, 24, 10, 14, 2, 26, 20, 27, 19, 0, 28, 8, 0, 29, 8, 16, 15, 25, 5, 19, 29, 9, 4, 19, 31, 16, 9, 10, 15, 19, 15, 25, 1, 15
, 3, 17, 26, 9, 35, 15, 11, 13, 25, 1, 19, 19, 8, 15, 4, 25, 15, 19, 6, 11, 29, 5, 1, 21, 19, 2, 21, 19, 13, 26, 15, 9, 28, 15, 28, 21, 16, 10, 4, 6, 43, 18, 7, 25, 25, 30, 3, 51, 3, 14, 5, 8, 16, 10, 20, 3, 24, 13, 12, 9, 18
, 19, 31, 1, 9, 3, 20, 15, 1, 15, 21, 24, 4, 21, 25, 26, 19, 15, 2, 5, 28, 11, 8, 22, 21, 9, 12, 21, 19, 13, 27, 22, 6, 8, 19, 7, 21, 18, 18, 14, 13, 11, 30, 29, 18, 8, 24, 8, 12, 4, 18, 9, 30, 30, 30, 30, 16, 15, 18, 0, 28,
1, 30, 2, 7, 29, 9, 12, 6, 20, 6, 10, 9, 29, 4, 10, 53, 20, 52, 3, 29, 1, 31, 8, 11, 24, 15, 7, 23, 3, 1, 31, 13, 30, 26, 11, 20, 23, 9, 7, 16, 25, 14, 9, 18, 4, 25, 9, 29, 0, 17, 3, 4, 21, 4, 6, 5, 26, 23, 6, 31, 31, 7, 14,
4, 18, 24, 41, 24, 23, 31, 1, 20, 26, 8, 27, 3, 1, 52, 17, 19, 23, 24, 28, 2, 25, 10, 15, 22, 29, 15, 18, 54, 22, 3, 26, 21, 0, 28, 30, 6, 24, 21, 15, 4, 22, 27, 5, 17, 17, 26, 19, 10, 7, 7, 5, 4, 12, 12, 4, 12, 26, 0, 28, 3,
 16, 23, 31, 26, 15, 15, 21, 30, 1, 26, 26, 2, 4, 2, 29, 6, 10, 3, 16, 25, 11, 27, 8, 22, 7, 12, 7, 2, 8, 22, 19, 17, 50, 6, 17, 29, 4, 10, 28, 0, 3, 5, 7, 12, 30, 9, 31, 39, 16, 5, 28, 6, 48, 5, 21, 8, 26, 5, 58, 7, 30, 30,
29, 25, 14, 22, 11, 8, 31, 27, 26, 21, 1, 13, 15, 2, 11, 3, 28, 0, 21, 26, 27, 20, 0, 24, 4, 8, 13, 2, 30, 26, 22, 16, 3, 27, 5, 25, 1, 1, 9, 14, 23, 24, 48, 8, 4, 23, 7, 28, 19, 39, 18, 16, 17, 29, 6, 16, 31, 5, 0, 32, 15, 2
0, 14, 0, 6, 10, 14, 6, 9, 31, 15, 20, 19, 19, 7, 21, 9, 23, 6, 17, 22, 25, 19, 22, 60, 1, 12, 15, 19, 17, 4, 13, 25, 14, 5, 6, 23, 6, 20, 27, 18, 24, 10, 2, 18, 26, 34, 21, 14, 8, 7, 16, 9, 4, 21, 8, 30, 16, 7, 19, 16, 21, 6
, 6, 5, 26, 31, 20, 21, 12, 8, 3, 2, 4, 19, 27, 12, 13, 13, 30, 0, 31, 16, 0, 14, 32, 28, 16, 21, 29, 5, 6, 26, 27, 2, 31, 12, 1, 22, 22, 11, 12, 18, 15, 18, 15, 10, 16, 45, 2, 19, 13, 24, 6, 22, 50, 13, 20, 16, 5, 6, 19, 52,
 28, 16, 6, 15, 13, 31, 18, 12, 19, 51, 28, 28, 49, 23, 2, 13, 28, 24, 28, 7, 24, 21, 21, 26, 19, 17, 7, 2, 5, 23, 16, 20, 22, 26, 15, 6, 5, 11, 29, 1, 0, 29, 30, 6, 20, 13, 30, 15, 16, 17, 29, 8, 41, 1, 22, 19, 0, 14, 13, 18
, 20, 9, 31, 17, 3, 2, 1, 16, 9, 3, 21, 27, 18, 3, 7, 29, 16, 21, 22, 23, 0, 10, 4, 10, 13, 28, 37, 27, 18, 24, 26, 41, 13, 9, 19, 20, 2, 16, 8, 25, 16, 2, 13, 1, 16, 24, 1, 22, 3, 31, 5, 0, 1, 0, 14, 22, 23, 8, 22, 5, 30, 31
, 42, 30, 25, 10, 30, 1, 12, 26, 1, 29, 3, 17, 15, 13, 8, 20, 4, 12, 2, 29, 24, 30, 0, 25, 2, 18, 31, 19, 10, 24, 31, 17, 4, 25, 20, 30, 28, 27, 51, 15, 11, 11, 17, 23, 1, 31, 9, 1, 29, 4, 18, 14, 31, 25, 8, 6, 7, 26, 3, 26,
28, 38, 2, 18, 26, 9], [16, 13, 2, 24, 4, 16, 13, 8, 29, 23, 0, 20, 12, 23, 22, 18, 27, 11, 26, 13, 22, 14, 32, 4, 12, 12, 3, 6, 11, 24, 10, 17, 26, 12, 29, 10, 25, 9, 10, 8, 3, 2, 26, 0, 15, 17, 1, 14, 20, 27, 19, 30, 26, 10
, 1, 12, 11, 17, 32, 15, 3, 15, 15, 7, 7, 4, 4, 11, 10, 7, 19, 26, 25, 4, 0, 25, 2, 12, 14, 6, 17, 14, 28, 25, 12, 24, 28, 14, 9, 17, 4, 23, 18, 11, 9, 9, 27, 25, 19, 17, 19, 10, 2, 20, 6, 3, 10, 8, 11, 22, 14, 0, 8, 26, 18,
16, 10, 10, 7, 26, 1, 25, 23, 6, 15, 24, 30, 5, 3, 1, 14, 12, 22, 13, 14, 10, 1, 25, 13, 23, 26, 28, 23, 6, 29, 18, 8, 26, 29, 12, 9, 5, 29, 14, 14, 5, 21, 4, 28, 2, 25, 0, 23, 30, 32, 16, 20, 25, 18, 17, 14, 32, 6, 5, 23, 27
, 21, 27, 17, 7, 19, 2, 13, 21, 16, 19, 26, 18, 8, 12, 24, 10, 4, 11, 23, 12, 18, 28, 21, 19, 0, 30, 4, 19, 28, 10, 0, 22, 2, 12, 14, 28, 28, 12, 7, 14, 8, 7, 28, 3, 28, 28, 2, 1, 9, 3, 30, 30, 29, 4, 23, 10, 3, 15, 1, 17, 21
, 16, 17, 19, 31, 8, 28, 15, 21, 0, 27, 10, 9, 24, 10, 3, 21, 12, 13, 4, 9, 14, 24, 24, 21, 12, 32, 17, 20, 12, 29, 7, 7, 23, 13, 22, 18, 8, 14, 18, 8, 16, 30, 4, 22, 23, 7, 8, 25, 11, 30, 6, 30, 25, 17, 6, 31, 0, 2, 30, 11,
17, 13, 8, 12, 22, 30, 2, 29, 31, 12, 2, 27, 10, 24, 32, 23, 19, 17, 21, 31, 26, 25, 8, 10, 15, 32, 2, 11, 28, 11, 25, 9, 11, 25, 1, 22, 14, 19, 0, 22, 14, 23, 8, 28, 13, 30, 17, 10, 23, 23, 17, 23, 25, 9, 14, 30, 29, 7, 24,
5, 23, 22, 12, 9, 9, 29, 10, 3, 8, 12, 19, 1, 18, 7, 14, 8, 28, 15, 6, 23, 19, 9, 19, 9, 4, 18, 18, 13, 10, 11, 12, 27, 11, 30, 27, 30, 23, 31, 31, 17, 7, 5, 15, 3, 18, 30, 17, 31, 15, 6, 3, 24, 3, 0, 20, 14, 12, 12, 30, 7, 2
, 22, 19, 22, 16, 22, 8, 12, 31, 23, 26, 20, 23, 18, 20, 29, 12, 19, 25, 27, 22, 9, 22, 26, 3, 31, 30, 7, 15, 13, 25, 10, 23, 31, 23, 29, 24, 17, 10, 20, 30, 26, 21, 20, 25, 24, 19, 8, 0, 11, 13, 8, 1, 2, 24, 12, 32, 7, 23, 6
, 2, 29, 31, 7, 1, 13, 1, 12, 3, 20, 19, 10, 3, 32, 2, 29, 6, 16, 21, 24, 1, 25, 1, 31, 2, 29, 26, 23, 22, 1, 10, 30, 11, 24, 14, 25, 1, 5, 28, 12, 5, 30, 22, 9, 13, 3, 25, 1, 1, 15, 25, 15, 24, 18, 22, 26, 10, 16, 27, 23, 32
, 26, 2, 8, 15, 0, 26, 23, 26, 3, 17, 28, 14, 11, 21, 7, 21, 4, 10, 8, 19, 27, 23, 2, 13, 10, 10, 9, 22, 21, 2, 2, 15, 17, 8, 14, 16, 18, 29, 32, 25, 3, 6, 5, 14, 4, 17, 20, 18, 22, 29, 1, 28, 5, 14, 29, 0, 4, 29, 5, 14, 21,
13, 17, 22, 1, 5, 8, 25, 32, 30, 29, 0, 20, 15, 29, 20, 15, 31, 20, 23, 4, 12, 18, 31, 14, 32, 0, 14, 17, 15, 11, 31, 12, 9, 19, 6, 29, 25, 21, 23, 26, 24, 11, 11, 4, 18, 24, 7, 14, 17, 27, 30, 0, 24, 13, 16, 15, 28, 27, 10,
27, 15, 10, 13, 21, 29, 20, 16, 32, 3, 22, 22, 19, 1, 12, 9, 17, 4, 12, 20, 8, 16, 18, 16, 25, 18, 28, 22, 23, 18, 9, 18, 30, 21, 4, 4, 15, 3, 8, 3, 9, 30, 32, 9, 1, 17, 8, 32, 0, 11, 28, 24, 13, 18, 1, 20, 21, 12, 3, 14, 2,
11, 14, 26, 30, 20, 10, 17, 1, 8, 3, 10, 6, 8, 30, 1, 27, 19, 10, 27, 20, 24, 18, 14, 14, 26, 19, 31, 12, 16, 0, 9, 6, 8, 9, 28, 19, 18, 10, 13, 17, 25, 12, 0, 12, 17, 5, 21, 12, 9, 15, 11, 12, 23, 15, 27, 32, 18, 27, 26, 30,
 22, 4, 19, 21, 9, 29, 15, 18, 6, 7, 16, 3, 15, 9, 31, 15, 5, 29, 6, 17, 26, 10, 26])
Расшифрованный текст:
вотпримерстатьинатысячусимволов.этодостаточномаленькийтекст,оптимальноподходящийдлякарточектовароввинтернетилимагазинахилидлянебольшихинформационныхпубликаций.втакомтекстередкобываетболеедвухилитрёхабзацевиобычноодинподзаголо
вок.номожноибезнего.натысячусимволоврекомендованоиспользоватьодинилидваключаиоднукартину.текстнатысячусимволовэтосколькопримернослов?статистикапоказывает,чтотысячавключаетвсебястопятьдесятилидвестисловсреднейвеличины.но,еслиз
лоупотреблятьпредлогами,союзамиидругимичастямиречинаодинилидвасимвола,токоличествословнеизменновозрастает.вкопирайтерскойдеятельностипринятосчитатьтысячиспробеламиилибез.учетпробеловувеличиваетобъемтекстапримернонастоилидвест
исимволовименностолькоразмыразделяемсловасвободнымпространством.считатьпробелызаказчикинелюбят,
```