

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
Г: ПОТОЧНЫЕ ШИФРЫ	5
15. А5 /1	5

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)
```

Г: ПОТОЧНЫЕ ШИФРЫ

15. A5 /1

A5 — это поточный алгоритм шифрования, используемый для обеспечения конфиденциальности передаваемых данных между телефоном и базовой станцией в европейской системе мобильной цифровой связи GSM (Groupe Spécial Mobile).

Шифр основан на побитовом сложении по модулю два (булева операция «исключающее или») генерируемой псевдослучайной последовательности и шифруемой информации. В A5 псевдослучайная последовательность реализуется на основе трёх линейных регистров сдвига с обратной связью. Регистры имеют длины 19, 22 и 23 бита соответственно. Сдвигами управляет специальная схема, организующая на каждом шаге смещение как минимум двух регистров, что приводит к их неравномерному движению. Последовательность формируется путём операции «исключающее или» над выходными битами регистров.

Код программы:

```
# -*- coding:utf-8 -*-
from base import alphabet, input_for_cipher_short, input_for_cipher_long, out
put_from_decrypted
import re
import copy
reg_x_length = 19
reg_y_length = 22
reg_z_length = 23
key_one = ""
reg_x = []
reg_y = []
reg_z = []

def loading_registers(key):
    i = 0
    while(i < reg_x_length):
        reg_x.insert(i, int(key[i]))
        i = i + 1
    j = 0
    p = reg_x_length
    while(j < reg_y_length):
        reg_y.insert(j, int(key[p]))
        p = p + 1
        j = j + 1
    k = reg_y_length + reg_x_length
    r = 0
    while(r < reg_z_length):
```

```

        reg_z.insert(r, int(key[k]))
        k = k + 1
        r = r + 1

def set_key(key):
    if(len(key) == 64 and re.match("^[01]+$", key)):
        key_one = key
        loading_registers(key)
        return True
    return False

def get_key():
    return key_one

def to_binary(plain):
    s = ""
    i = 0
    for i in plain:
        binary = str(' '.join(format(ord(x), 'b') for x in i))
        j = len(binary)
        while(j < 12):
            binary = "0" + binary
            s = s + binary
            j = j + 1
    binary_values = []
    k = 0
    while(k < len(s)):
        binary_values.insert(k, int(s[k]))
        k = k + 1
    return binary_values

def get_majority(x, y, z):
    if(x + y + z > 1):
        return 1
    else:
        return 0

def get_keystream(length):
    reg_x_temp = copy.deepcopy(reg_x)
    reg_y_temp = copy.deepcopy(reg_y)
    reg_z_temp = copy.deepcopy(reg_z)
    keystream = []
    i = 0
    while i < length:
        majority = get_majority(reg_x_temp[8], reg_y_temp[10], reg_z_temp[10]
)

```

```

        if reg_x_temp[8] == majority:
            new = reg_x_temp[13] ^ reg_x_temp[16] ^ reg_x_temp[17] ^ reg_x_temp[18]

            reg_x_temp_two = copy.deepcopy(reg_x_temp)
            j = 1
            while(j < len(reg_x_temp)):
                reg_x_temp[j] = reg_x_temp_two[j-1]
                j = j + 1
            reg_x_temp[0] = new

        if reg_y_temp[10] == majority:
            new_one = reg_y_temp[20] ^ reg_y_temp[21]
            reg_y_temp_two = copy.deepcopy(reg_y_temp)
            k = 1
            while(k < len(reg_y_temp)):
                reg_y_temp[k] = reg_y_temp_two[k-1]
                k = k + 1
            reg_y_temp[0] = new_one

        if reg_z_temp[10] == majority:
            new_two = reg_z_temp[7] ^ reg_z_temp[20] ^ reg_z_temp[21] ^ reg_z_temp[22]

            reg_z_temp_two = copy.deepcopy(reg_z_temp)
            m = 1
            while(m < len(reg_z_temp)):
                reg_z_temp[m] = reg_z_temp_two[m-1]
                m = m + 1
            reg_z_temp[0] = new_two

        keystream.insert(i, reg_x_temp[18] ^ reg_y_temp[21] ^ reg_z_temp[22])
        i = i + 1
    return keystream
def convert_binary_to_str(binary):
    s = ""
    length = len(binary) - 12
    i = 0
    while(i <= length):
        s = s + chr(int(binary[i:i+12], 2))
        i = i + 12
    return str(s)
def encrypt(plain):
    s = ""
    binary = to_binary(plain)
    keystream = get_keystream(len(binary))
    i = 0
    while(i < len(binary)):
        s = s + str(binary[i] ^ keystream[i])
        i = i + 1
    return s

```

```

def decrypt(cipher):
    s = ""
    binary = []
    keystream = get_keystream(len(cipher))
    i = 0
    while(i < len(cipher)):
        binary.insert(i, int(cipher[i]))
        s = s + str(binary[i] ^ keystream[i])
        i = i + 1
    return convert_binary_to_str(str(s))

def user_input_key():
    tha_key = str(input('Введите 64-bit ключ: '))
    if (len(tha_key) == 64 and re.match("^[01]+$", tha_key)):
        return tha_key
    else:
        while(len(tha_key) != 64 and not re.match("^[01]+$", tha_key)):
            if (len(tha_key) == 64 and re.match("^[01]+$", tha_key)):
                return tha_key
            tha_key = str(input('Введите 64-bit ключ: '))
    return tha_key

# 01010010000110101100011100011001001001000000110111111010110111

key = str(user_input_key())
set_key(key)

print(f'''
A5/1:
КОРОТКИЙ ТЕКСТ:
Зашифрованный текст:
{encrypt(input_for_cipher_short())}

Расшифрованный текст:
{output_from_decrypted(decrypt(encrypt(
    input_for_cipher_short())))}

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

Расшифрованный текст:
{output_from_decrypted(decrypt(encrypt(
    input_for_cipher_long())))}
''')

```


Тестирование:

```
/bin/python3 /root/mospolytech-education-crypt-dev-2021-1/lab06_15_a51.py
```

Введите 64-bit ключ:

```
0101001000011010110001110001100100101001000000110111111010110111
```

A5/1:

КОРОТКИЙ ТЕКСТ:

Зашифрованный текст:

```
11011011101011101010000111010111001000001110010101111100100111011001001010110
00001011011100001111101111010001101101100110111000001000100110001011100111111
00110001110110110111110101101011100011111101001110011000001001101010010111101
10010001100110101001011101001010010001000100001010110010100101111101100101011
11100010001010010100111111011000011000000101000010011010011000100001010101111
00110000100110110111000011011100100001000010001000011010111011101100111101011
111000
```

Расшифрованный текст:

время, приливьиотливынеждутчеловека.

ДЛИННЫЙ ТЕКСТ:

Зашифрованный текст:

```
11011011101011101101111111010000010100001110011001111100011011011001110110110
00001101011111100001101100101111101101100100111011111100100110011101100100001
10110000001000110110000110101011100110111101000000011000001001101011100111101
10101101000110010111111101001100010001111011001010001000100101111010100101011
01100010010110000100100000011000000111000101000000101010011010110001001010111
00111110001110110110111011010010011001001100011000010101011011100010100101011
11110010011011100000010100110110001111101101111001110111111010110100101110100
01010101110101101000111001100001101110110011100100100011111010100001011011010
00111100011011100001111110100111000101101000101001110010101100011011111110001
11100110001011000101100000010110110001001110010000111001001001011101110101010
00110011101101110000000101001110011101010100111100001011000001001111011000001
00010001111111111110011010000110000000001101110000000010110111100001011011111
00011100100100100101100011101100010000000010101110011010101011111111011110010
11001110100110000100101111000001101010100100101101100001010101001000101011101
0001001000000000101001101011001011010111010111010101101000001101111100011
00010110011000111011011110000001000000001100110100011110011001011111011110110
11111000110000100111000000110101110110111000101011001101001001011110011100001
11010111010110001001011011111101011100011110000101000001010000011000100010101
00110010000010010111100011110000000001010000000101100111011100011101101010100
11001100111001110111001010001010010100110110010110001100011011100011101101101
000000101010001111101001001100110011001100110110100001010110110111100100100
0001100001001110110010011100001000100110111100001100100100000000010111101010
11100010001101011110010111001100011010110001111000110000000100011110101100011
00111001111100000010011101001000101011001100101000010010010111010010101011000
10011101100010001011111011111111011000001010010000011100000101000011000101101
101001000111001111111111010110000110001111100000111001010111011010111111101001
11011010001001111110000001010101011100000011011111010110000110111101100110100
00010011000001111100101011010111100000111011101100000010011110100101000010110
```

01010110010100011001111110100111101010110011101111011101011100101011000111000
00011010011011100110101000110111000101000100001001001100010000011100111111011
01100101101010000111001101100110100100000011110000010000000100101011001100100
11111100011110011011010100001000010010100100011101000101010010110010101111100
01011011100100010110011100100011101110101101010011001101110111010000111100000
01101010011010010110111001010110100000000010110100010011001011101110101010010
1110100101010110011101011011000001101001101101010111010110011111110000010011
01011111000101101111010010111010011111111001111010111100111101001110101010000
00001101110011100011001001010000111111001001101110000110001110001000000010000
00001100010110101111101100000000010100000010101110000010101001110011100101000
10011011000011111000000011110110100000101001101100111001000010110110111001101
11101100001110011001111110001010110001010110101100011011000101011011010110011
11110101000001011110111011111010011011110110110001001101001100110010010100111
01000110101010101110100101100010011111011010101001111010100111100010001001110
11001010001101100101100011011110011110001011001001000100011010010111111010011
10010011110001100011000101011111010111101101001101001011011100001111110111110
11001101000011001001001100101010011101110010011111101001000100110110011110010
10010101000000010110101101011100111011011010000100000010000001101011101000111
11111101001011101011110111100001110000001000111111000100000001101000001010100
01101000011110010001000111100011000111001000111110010010000101001111011100011
11100001010001011111101011101100010101100101111101011010101101100111000000111
10100001010111101000110010100111000000100110111101101111000111010010101110111
10110000111010101010110001111000000010100100110010101110001101101110000000000
1010011110000100101110010101011111111010001110100001100110111010101110111000
11101010111010101101111100010101011001100011010111011111010110110110110110
10011100000001111111101000101010010110001011111000010110111000010011001010010
11010011101100001001001001001111011001011010010000100111110010100110111101001
10111010101011000001000010111000010110010000000010001101011100000111000110110
11000000110001100011001110010001000000100011111001100011000011011010011110000
00000100100101110000010100011011000001000110001100100001001100001110010110001
10101101110100011100110110001100111000010010010110101101101010110011100111110
00101100011000110000101101001011100111100010001101001110011010110111100110001
00011111001100100010100110001010110101111110001001100100010000111000011111101
00110111110010010010000111100010000110001001001100000010011110011101110011100
11011001010100000010101001110000110010111000001001110111000001100100100000001
01001111101110010100011000011000010001100110101000011111100011110100010010111
01101000000010000111101111100110000101100001001001100111000010011001010011100
00111111101011000001110100000100101101101010001000111111001100010011011100110
10101111011101000011111011111011101010110000011000111000000111000100100110100
10111110111001110100101001011101011111100011011111101101101111110000001001
01011010111110111010000110101010100001100101111011001000000111100110000011111
1011011111001110001110000011001101100100101111000000001000011100110001101101
10001101011010110111000000011100000001101110101101110001011110010100001010100
10010000000111001001101010101000101011001000010110011111010001110100110011010
11011111100011000001001010011011111010001000001010011101011001111100000111111
00010111100101001010110111001111000000100001011001010001000100010110110101
11010110110111010010110101100001100111101011010000101000000000100101100111100
10110101110000101111101001101100000010100111111011100100010000000111100000101
11101001110111010001011010011010011101010100111101111000001000011001001111111
01101010110010101001110001110111000010011011000110011000011101000001000110111

1100100101010010001111111101100101111011011000001101110011101110100111100011
01101110111001010110101011001100110100000100101110010101011011101010111011011
0111100100101111010001010101111001100101001001011100011101011000001111111111
00111110010000010000111110100011000001011100011100100101001001001010111110111
01110110010110101011011001011000010010111111010000100011111000001101101100000
10011111100001010001010000101010100011110111000001110000100101101010111100010
11100001001001000001100000111101111110001111111001101110101001111111101010110
11101011001100111001101000100111101111000100100000100100111100001010111001010
11011100111000000110010010100011000010101101110011101011111010000000010010001
11101110100100011011011100111100100001101110100100110011010000011101001101011
00000001110111111000110110100100100110000100001101010011100001011101000110110
00000100110001000111011001101100010100111111100000000010110111011010011101110
00101010001000011110001101010000010010111011000100011001010111110010110100001
10100100110001110110000100011000010011011000001000001101010011000001100100100
11111011011100111011001001000111111011001011101101011100100100111110010001001
10111011010010001111000010001110110011100000001101100001111000111001101000000
10110101000001100111101001011101000011101111111010001010101100111011111011001
11000110001001101001001000000000001110010011111010110000011010010010100110011
11100000010101000110110011110110001011101011101100011000100101100011001111010
00100110100011011010110010001011111010001010010101000110111110110110000110100
10000011111110011000100100101000000100101000011100111100001110110011111001000
11110011000000101000110000101101110011001110110011010000000010010100111101110
11111100001001000110000110111100010000111101101000101001011010100001010010011
10001011001111000101001100010000010100001011000101100101100101011000001110100
00110100110001101111010000000011101110000101000110101011010100111110000010111
00100111101100001000100111100011000100101001111001011000010110110000110010100
111101101001110000001111001001110010010101000001111010100101001111010101001100
1111111000010001101011011000001101000101001100110111101101010100111010101100
00010000111110000010011100001010100100000001110001011011100111101000010010011
11011111010101110001001010111111001110001100110010110100111000001110010000111
01100110010111100100000011101111001100001100101010100000001011001010000101101
01011110101010100011111110111011010010000100111101111000111000111001110101001
10101110011101101101111111001110011010100001010101110100011110111011100000110
01101011000000111010001100110011011101010001010101100101011010111000000001000
10011100100011110100011110000100001000100001111001010110100011001100100010100
1000001010100101110111111111011011110110001101010010011110101010101001011100
01101111010011000111010110000011011100000010011110101111110100100001100011001
111100000101011011001100101001000101010101110111101101001010000101110101011
10101101001000001000000110101100000101101100011000000001001011101101001001111
11100101101100110001001110101101000000111100010110000010101000011110010011110
1101001000000000111000001100001101001000011011111010101100111110111010111111
1101100111110001011101000001011111110000011000001001111110101000001100110111
1010111011010100110001111001111110111000010110000110010011000000000111101111
10010100101010110011100100100001100110010111110011100110111110000000110000010
00001001000010001111101011100011100100110101101110001000100101100100110000000
00100011111110000111001100111111111101001011011001010000001010100010000010101
00010000100111001001101101110000100110101101110001000100101100100110000000
00011101000101100000000011001001111111101001111100100001110001010000010110101
11101010001000011111101001101010111010000100101010011001010000000110001110000
1110011001101011010000000101000101000100001011110110111011010001101001011100

10010110111011010010001110110001111100100010011111110000000110000011000010111
00100010011100100010000001101110101111010110110011010011011101111000011111001
00011000001000010101010111001110011110101111101001011010001101110111111110010
00110001000100111001100111010111100011001101001101101101101001001101001111
01111110000010100100111000010011010111001101000111111011000001111010101010100
00111010111111001100000000010101010110101100111001001110010011011000101011101
01110100000010010001100100111000111011100001111000100111111001110000001000010
00011100111000101000101000011100111011011100000010011010110011100111001111010
10111010011011010000010101010000111000100100101000000110110111110101011011001
01100001010011110000110011000111111010000100100111000111101100110101010010100
00011100001010010100111010101000001101110011000001011010001001100010100011000
1111110100101111010000100100010001110010110101000111011111110000110111001001
11111110101010111100000010110000111011100001010011010101000000010011001101101
10101000110110001011101000111100100000010001000011010111011111110010010001101
00011001111101101011100110000011101101110111101000101101100111110001101111001
00111011011100000001000011000010000011001001111010001111001001011011111111001
0110100101011111000100001010101000100000010100001011011101111111110110111001
1000000000101110000011110011111111111101011101110100100011110001100101010100
01101110001011011111001010011001101110011001111000110011001011001000100001011
0110111111000000001000110111100001100010011111110000010001001011100101000000
00010101000000101111011001011110110100010101100111001000100100000111111001000
10001111011001001000011110100001000001011001010111101100011011110010100011100
1001111100001100100011101100000011011010111111101011111010111111001100101101
11011001110111001101111000010000010101010001101000001111011110010110001000111
1100100011100100000011001110110101010011101010011000100110100101000001011111
000111000010100010101011101001011000001010011001001111011110101010001101001
01100101101100101001100101000000000101110011101010101111011001101010010010000
1010110001001011101011011010011010000001101010000011100100000100001001011001
1011000111000100110110010111110110000001011111111110001001011001100100111111
01011110100001100100100110001110100110011001111111100101011111101100000110001
000010001001101000111100101011010111000001001010001110000111001000011110111
000111010110011110100001010001010110001001000111101111001000101100100101100
11011000011000000100011100010110011010011001100001111011010001010011000100101
11111100010010100000110110100111100011000111001011101010010101110001010110011
11111110100011011100001110101001011000110111001111011100000100100100110101001
11111010010001111000100000101101111101110010001010000011000111101001010101001
11001001000101111000100101100011001011101111001110100101010000001000000100010
10110101101110100101100000011111101011100010010111000111011000010101100100001
01011001100000011011110011001111000001001100111111100110111001000110111100110
110101100100111010100011011001110001011110101000001110010001001011011111000110
00100110110100010011000100000110100111011101011101101011101110100100110011000
0110010011111101010101011011010

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 рекомендуется использовать один или два ключа и одну картинку. текст на тысячу символов это сколько примерно слов. статистика показывает, что тысяча включает в себя пятьдесят или двести слов средней величины. но, если злоупотреблять

ь предложениями, союзами и другими частями речи на один или два символа, то количество слов не изменяется. В копирайтерской деятельности принято считать тысячу 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 биржи видят справедливым ставить стоимость за тысячу символов пробелами, считая последние важным элементом качественного восприятия. Согласитесь, читать слитный текст без единого пропуска, никто не будет. Но больше никому не нужна цена за тысячу знаков без пробелов.