## 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#

## ГЕНЕРАЦИЯ ЦИФРОВОЙ ПОДПИСИ

- **ГОСТ Р 34.10-94**

**Код программы:**

```
alphavit = {'а': 0, 'б': 1, 'в': 2, 'г': 3, 'д': 4,
            'е': 5, 'ё': 6, 'ж': 7, 'з': 8, 'и': 9, 'й': 10,
            'к': 11, 'л': 12, 'м': 13, 'н': 14, 'о': 15,
            'п': 16, 'р': 17, 'с': 18, 'т': 19, 'у': 20,
            'ф': 21, 'х': 22, 'ц': 23, 'ч': 24, 'ш': 25,
            'щ': 26, 'ъ': 27, 'ы': 28, 'ь': 29, 'э': 30,
            'ю': 31, 'я': 32
            }


def ciphergostd(clearText):
    array = []
    flag = False
    for s in range(50, 1000):
        for i in range(2, s):
            if s % i == 0:
                flag = True
                break
        if flag == False:
            array.append(s)
        flag = False
    p = 31
    print("p = ", p)
    q = 5
    print("q = ", q)
    a = 2
    print("a =", a)

    array2 = []
    flag2 = False
    for s in range(2, q):
        for i in range(2, s):
            if s % i == 0:
                flag2 = True
                break
        if flag2 == False:
            array2.append(s)
        flag2 = False

    x = 3
    print("x = ", x)
    y = a**x % p
    k = 4
    print("k = ", k)
```

```
    r = (a**k % p) % q

    msg = clearText
    msg_list = list(msg)
    alpha_code_msg = list()
    for i in range(len(msg_list)):
        alpha_code_msg.append(int(alphavit.get(msg_list[i])))
    print("Длина исходного сообщения {} символов".format(len(alpha_code_msg)))
    hash_code_msg = hash_value(p, alpha_code_msg)
    print("Хэш сообщения:= {}".format(hash_code_msg))

    s = (x*r+k*hash_code_msg) % q

    print("Цифровая подпись = ", r % (2**256), ",", s % (2**256))

    v = (hash_code_msg**(q-2)) % q
    z1 = s*v % q
    z2 = ((q-r)*v) % q
    u = (((a**z1)*(y**z2)) % p) % q
    print(r, " = ", u)
    if u == r:
        print("r = u, следовательно:")
        print("Подпись верна\n")
    else:
        print("Подпись неверна")


def hash_value(n, alpha_code):
    i = 0
    hash = 1
    while i < len(alpha_code):
        hash = (((hash-1) + int(alpha_code[i]))**2) % n
        i += 1
    return hash

def main():
    print('ГОСТ Р 34.10-94:')
    message = input("Введите сообщение: ")
    ciphergostd(message)


if __name__ == "__main__":
    main()
```

**Тестирование:**

**Фраза по варианту**

```
PS C:\Users\xiaomi\Desktop\cryptography_ciphers\lab_10\gost_94> python3 .\gost.py
ГОСТ Р 34.10-94:
Введите сообщение: красивымисловамипастернакнепомаслишь
p =  31
q =  5
a = 2
x =  3
k =  4
Длина исходного сообщения 36 символов
Хэш сообщения:= 5
Цифровая подпись =  1 , 3
1  =  1
r = u, следовательно:
Подпись верна
```

## Текст на 1000 символов

```
PS C:\Users\xiaomi\Desktop\cryptography_ciphers\lab_10\gost_94> python3 .\demo.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 информаци
онных публикаций. В таком тексте редко бывает более двух или трёх абзацев и обычно один подзаголовок. Но можно и без него. На тысячу символов рекомендовано использов
ать один или два ключа и одну картину. Текст на тысячу символов это сколько примерно слов? Статистика показывает, что тысяча включает в себя стопятьдесят 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 копирай
терской деятельности принято считать тысячи с пробелами или без. Учет пробелов увеличивает объем текста примерно на сто или двести символов именно столько раз мы раз
деляем слова свободным пространством. Считать пробелы заказчики не любят, так как это пустое место. Однако некоторые фирмы и биржи видят справедливым ставить стоимос
ть 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.
p =  31
q =  5
a = 2
x =  3
k =  4
Длина исходного сообщения 1087 символов
Хэш сообщения:= 20
Цифровая подпись =  1 , 3
1  =  1
r = u, следовательно:
Подпись верна
```

- **ГОСТ Р 34.10-2012**

## Программа

```python
import random
import collections

alphabet_lower = {'а': 0, 'б': 1, 'в': 2, 'г': 3, 'д': 4,
            'е': 5, 'ё': 6, 'ж': 7, 'з': 8, 'и': 9, 'й': 10,
            'к': 11, 'л': 12, 'м': 13, 'н': 14, 'о': 15,
            'п': 16, 'р': 17, 'с': 18, 'т': 19, 'у': 20,
            'ф': 21, 'х': 22, 'ц': 23, 'ч': 24, 'ш': 25,
            'щ': 26, 'ъ': 27, 'ы': 28, 'ь': 29, 'э': 30,
            'ю': 31, 'я': 32
            }


class Point:
    def __init__(self, x_init, y_init):
        self.x = x_init
        self.y = y_init

    def shift(self, x, y):
        self.x += x
```

```
        self.y += y

    def __repr__(self):
        return "".join(["( x=", str(self.x), ", y=", str(self.y), ")"])


x_1 = 0
y_1 = 0

EllipticCurve = collections.namedtuple(
    'EllipticCurve', 'name p q_mod a b q g n h')
curve = EllipticCurve(
    'secp256k1',
    p=0xfffffffffffffffffffffffffffffffffffffffffffffffffffffffefffffc2f,
    q_mod=0xfffffffffeffffffffcfffffffffffcfffffffffffffffffffffffefffffc2f,

    a=7,
    b=11,

    g=(0x79be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798,
       0x483ada7726a3c4655da4fbfc0e1108a8fd17b448a68554199c47d08ffb10d4b8),
    q=(0xA0434D9E47F3C86235477C7B1AE6AE5D3442D49B1943C2B752A68E2A47E247C7,
       0x893ABA425419BC27A3B6C7E693A24C696F794C2ED877A1593CBEE53B037368D7),

    n=0xfffffffffffffffffffffffffffffffebaaedce6af48a03bbfd25e8cd0364141,

    h=1,
)


def ciphergostd(clearText):
    msg = clearText
    msg_list = list(msg)
    alpha_code_msg = list()
    for i in range(len(msg_list)):
        alpha_code_msg.append(int(alphabet_lower.get(msg_list[i])))
    print("Длина исходного сообщения {} символов".format(len(alpha_code_msg)))

    print("Q mod", int(curve.q_mod))
    print("P mod", int(curve.p))

    hash_code_msg = hash_value(curve.p, alpha_code_msg)
    print("Хэш сообщения:={}".format(hash_code_msg))

    e = hash_code_msg % curve.q_mod
    print("E={}".format(e))

    k = random.randint(1, curve.q_mod)
    print("K={}".format(k))
```

```
    d = 10
    print("D={}".format(d))
    x, y = scalar_mult(k, curve.g)
    point_c = Point(x, y)
    print("Point_C={}".format(point_c))
    r = point_c.x % curve.q_mod
    print("R={}".format(r))
    s = (r*curve.p + k*e) % curve.q_mod
    print("S={}".format(s))

    v = inverse_mod(e, curve.p)
    print("V={}".format(v))
    z1 = (s*v) % curve.q_mod
    z2 = ((curve.p-r)*v) % curve.q_mod
    x_1, y_1 = scalar_mult(d, curve.g)
    print("Point_Q=( x={}, y={} )".format(x_1, y_1))
    point_c_new = Point(x, y)
    x, y = point_add(scalar_mult(z1, curve.g),
                     scalar_mult(z2, curve.q))
    r_1 = point_c_new.x % curve.q_mod
    print("R_new={}".format(r_1))
    if r == r_1:
        print("Подпись прошла проверку!\n")
    else:
        print("Ошибка проверки!")


def hash_value(mod, alpha_code_msg):
    i = 0
    hashing_value = 1
    while i < len(alpha_code_msg):
        hashing_value = (
            ((hashing_value-1) + int(alpha_code_msg[i]))**2) % curve.p
        i += 1
    return hashing_value


def is_on_curve(point):
    if point is None:
        return True

    x, y = point

    return (y * y - x * x * x - curve.a * x - curve.b) % curve.p == 0


def point_neg(point):
    if point is None:
        return None
    x, y = point
```

```
    result = (x, -y % curve.p)
    return result


def inverse_mod(k, p):
    if k == 0:
        raise ZeroDivisionError('деление на 0')

    if k < 0:
        return p - inverse_mod(-k, p)

    s, old_s = 0, 1
    t, old_t = 1, 0
    r, old_r = p, k

    while r != 0:
        quotient = old_r // r
        old_r, r = r, old_r - quotient * r
        old_s, s = s, old_s - quotient * s
        old_t, t = t, old_t - quotient * t

    gcd, x, y = old_r, old_s, old_t

    assert gcd == 1
    assert (k * x) % p == 1

    return x % p


def point_add(point1, point2):
    if point1 is None:
        return point2
    if point2 is None:
        return point1

    x1, y1 = point1
    x2, y2 = point2

    if x1 == x2 and y1 != y2:
        return None

    if x1 == x2:
        m = (3 * x1 * x1 + curve.a) * inverse_mod(2 * y1, curve.p)
    else:
        m = (y1 - y2) * inverse_mod(x1 - x2, curve.p)

    x3 = m * m - x1 - x2
    y3 = y1 + m * (x3 - x1)
    result = (x3 % curve.p,
              -y3 % curve.p)
```

```
    return result


def scalar_mult(k, point):
    if k % curve.n == 0 or point is None:
        return None

    if k < 0:
        return scalar_mult(-k, point_neg(point))

    result = None
    addend = point

    while k:
        if k & 1:
            result = point_add(result, addend)
        addend = point_add(addend, addend)
        k >>= 1
    return result


def main():
    print('ГОСТ Р 34.10-2012:')

if __name__ == "__main__":
    main()
```

## Тестирование

## Фраза по варианту

```
PS C:\Users\xiaomi\Desktop\cryptography_ciphers\lab_10\gost_2012> python3 .\demo.py
Введите текст
Красивыми словами пастернак не помаслишь
Длина исходного сообщения 36 символов
Q mod 115792089237210883131902140479076077470404524942491262870694982560773809634351
P mod 115792089237316195423570985008687907853269984665640564039457584007908834671663
Хэш сообщения:=41734968009033504843595667972719937160002463522949733803928547400276299376185
E=41734968009033504843595667972719937160002463522949733803928547400276299376185
K=5410149137837268216218828610614337238946429548169683365750331746119086108448 5
D=10
Point_C=( x=33546030276856489284313896481160760174851777093607859709967439619313428716642, y=369451525768681177800000903791635480779893439661162303426012330480200971
24338)
R=33546030276856489284313896481160760174851777093607859709967439619313428716642
S=33911622774720931660243986213563325266950198836687131218874451305856193136938
V=48547999535034560835132157881527605612194084907292994747664768790559022707003
Point_Q=( x=10980558621116620662943286689258323111755451026059660014288829012550799306711 8, y=5124308323550405832119153432373625082229744368175398411412115647493855 0647252 )
R_new=33546030276856489284313896481160760174851777093607859709967439619313428716642
Подпись прошла проверку!
```

## Проверка текста на 1000 символов

```
PS C:\Users\xiaomi\Desktop\cryptography_ciphers\lab_10\gost_2012> python3 .\demo.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 информаци
онных публикаций. В таком тексте редко бывает более двух или трёх абзацев и обычно один подзаголовок. Но можно и без него. На тысячу символов рекомендовано использов
ать один или два ключа и одну картину. Текст на тысячу символов это сколько примерно слов? Статистика показывает, что тысяча включает в себя стопятьдесят 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 копирай
терской деятельности принято считать тысячи с пробелами или без. Учет пробелов увеличивает объем текста примерно на сто или двести символов именно столько раз мы раз
деляем слова свободным пространством. Считать пробелы заказчики не любят, так как это пустое место. Однако некоторые фирмы и биржи видят справедливым ставить стоимос
ть 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.
Длина исходного сообщения 1087 символов
Q mod 115792089237210883131902140479076077470404524942491262870694982560773809634351
P mod 115792089237316195423570985008687907853269984665640564039457584007908834671663
Хэш сообщения:=34910327282876669981090751617174695655616842716415000931628825868664952301091
E=34910327282876669981090751617174695655616842716415000931628825868664952301091
K=51928834512219181248301315739408102453486096158365035370298806285380244488479
D=10
Point_C=( x=85092709815368021679044415151341490324671440174964840286300895809991344314208, y=80034903046451462732810697071604734601662379972768027431302461569251495220177)
R=85092709815368021679044415151341490324671440174964840286300895809991344314208
S=65542983399663482991201482682038200562939820423034580932737216706138149533788
V=80771434365083502859973299123302845370380616728395900081312955467152482216320
Point_Q=( x=109805586211166206629432866892583231117554510260596600142888290125507993067118, y=51243083235504058321191534323736250822297443681753984114121156474938550647252 )
R_new=85092709815368021679044415151341490324671440174964840286300895809991344314208
Подпись прошла проверку!
```