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

- **Elgamal**

Код программы:

```
from math import gcd
import random
#инициализация алфавита
alphavit = {'а':0, 'б':1, 'в':2, 'г':3, 'д':4,
                  'е':5, 'ж':6, 'з':7, 'и':8, 'й':9,
                  'к':10, 'л':11, 'м':12, 'н':13, 'о':14,
                  'п':15, 'р':16, 'с':17, 'т':18, 'у':19,
                  'ф':20, 'х':21, 'ц':22, 'ч':23, 'ш':24,
                  'щ':25, 'ъ':26, 'ы':27, 'ь':28, 'э':29,
                  'ю':30, 'я':31, ' ':32, ",":33, ".":34
                  }

#проверка на простое число
def IsPrime(n):
    d = 2
    while n % d != 0:
        d += 1
    return d == n

#расширенный алгоритм Евклида или (e**-1) mod fe
def modInverse(e,el):
    e = e % el
    for x in range(1,el):
        if ((e * x) % el == 1):
            return x
    return 1

#выбор простого целого P, выбор целого числа G,G<P
def is_prime(num, test_count):
    if num == 1:
        return False
    if test_count >= num:
        test_count = num - 1
    for x in range(test_count):
        val = random.randint(1, num - 1)
        if pow(val, num-1, num) != 1:
            return False
    return True

def gen_prime(n):
    found_prime = False
    while not found_prime:
        p = random.randint(2**(n-1), 2**n)
        if is_prime(p, 1000):
```

```
            return p

p = gen_prime(10)
print("P =",p)
print()
g = random.randint(2,p-1)
print("G =",g)
print()
#отправитель выбирает случайное целое число X,1<x<(p-1)
x = random.randint(2,p-2)
y = (g**x)%p
print("Открытый ключ(Y)={}, Секретный ключ(X)={}".format(y,x))
print()
#хэшируем сообщение
msg = input("Введите сообщение:")
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)))
print()

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

hash_code_msg = hash_value(p, alpha_code_msg)
print("Хэш сообщения:= {}".format(hash_code_msg))
print()
#генерация случайное целое число K
k = 1
while True:
    k = random.randint(1,p-2)
    if gcd(k,p-1) == 1:
        print("K =",k)
        break

#отправитель вычисляет число целое число a
a = (g**k)%p
#вычисляем b
b = modInverse(k,p-1) * ((hash_code_msg - (x * a))%(p-1))
#b = modInverse((int(hash_code_msg) - int(x)*int(a)),p-1)
print("Значение подписи:S={},{}".format(a,b))
print()

#првоерка подписи (передвём m, a,b)
```

```
check_hash_value = hash_value(p, alpha_code_msg)
a_1 = ((y**a) * (a**b)) % p
print("A1={}".format(a_1))
print()
a_2 = (g**check_hash_value)%p
print("A2={}".format(a_2))
print()
if a_1 == a_2:
    print("Подпись верна")
else:
    print("Подпись неверна")
```

Тестирование:

Фраза по варианту

```
PS C:\Users\xiaomi\Desktop\cryptography_ciphers\lab_8\elgamal> python3 main.py
P = 727

G = 257

Открытый ключ(Y)=563, Секретный ключ(X)=718

Введите сообщение:красивыми словами пастернак не помаслишь
Длина исходного сообщения 40 символов

Хэш сообщения:= 482

K = 155
Значение подписи:S=715,35066

A1=451

A2=451

Подпись верна
```

Текст на 1000 символов

```
PS C:\Users\xiaomi\Desktop\cryptography_ciphers\lab_8\elgamal> python3 main.py
P = 863

G = 268

Открытый ключ(Y)=27, Секретный ключ(X)=240

Введите сообщение:вот пример статьи на тысячу символов. это достаточно маленький текст, оптимально подходящий для карточек товаров в интернет или магазинах или для н
ебольших информационных публикаций. в таком тексте редко бывает более двух или трех абзацев и обычно один подзаголовок. но можно и без него. на тысячу символов реком
ендовано использовать один или два ключа и одну картину. текст на тысячу символов это сколько примерно слов. сатистика показывает, что тысяча включает в себя стопять
десят или двести слов средней величины. но, если злоупотреблять предлогами, союзами и другими частями речи на один или два символа, то количество слов неизменно возр
астает. в копирайтерской деятельности принято считать тысячи с пробелами или без. учет пробелов увеличивает обьем текста примерно на сто или двести символов именно с
только раз мы разделяем слова свободным пространством. считать пробелы заказчики не любят, так как это пустое место. однако некоторые фирмы и биржи видят справедливы
м ставить стоимость за тысячу символов с пробелами, считая последние важным элементом качественного восприятия. согласитесь, читать слитный текст без единого пропуск
а, никто не будет. но большинству нужна цена за тысячу знаков без пробелов.
Длина исходного сообщения 1212 символов

Хэш сообщения:= 639

K = 743
Значение подписи:S=184,249165

A1=427

A2=427

Подпись верна
```

- **RSA**

Программа

```
from math import gcd
#инициализация алфавита
alphabet_lower = {'а':0, 'б':1, 'в':2, 'г':3, 'д':4,
                  'е':5, 'ж':6, 'з':7, 'и':8, 'й':9,
                  'к':10, 'л':11, 'м':12, 'н':13, 'о':14,
                  'п':15, 'р':16, 'с':17, 'т':18, 'у':19,
                  'ф':20, 'х':21, 'ц':22, 'ч':23, 'ш':24,
                  'щ':25, 'ъ':26, 'ы':27, 'ь':28, 'э':29,
                  'ю':30, 'я':31, ' ':32, ",":33, ".":34
                  }

#проверка на простое число
def IsPrime(n):
    d = 2
    while n % d != 0:
        d += 1
    return d == n
#расширенный алгоритм Евклида или (e**-1) mod fe
def modInverse(e,el):
    e = e % el
    for x in range(1,el):
        if ((e * x) % el == 1):
            return x
    return 1

#инициализация p,q,e,n
p = int(input("Введите p: "))
print(IsPrime(p))
q = int(input("Введите q: "))
print(IsPrime(q))
n = p * q
print("N =",n)
el = (p-1) * (q-1)
print("El =",el)
e = 257
print("E =",e)
if gcd(e,el) == 1:
    print(gcd(e,el),"E подходит")
else:
    print(gcd(e,el),"False")
#нахождение секретной экспоненты D
d = modInverse(e,el)
print("D =",d)
print("Открытый ключ e={} n={}".format(e,n))
print("Секретный ключ d={} n={}".format(d,n))
#хэширование сообщения
```

```python
msg = input("Введите сообщение:")
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)))
def hash_value(n,alpha_code):
    i = 0
    hashing_value = 1
    while i < len(alpha_code_msg):
        hashing_value = (((hashing_value-1) + int(alpha_code_msg[i]))**2) % n
        i += 1
    return hashing_value

hash_code_msg = hash_value(n, alpha_code_msg)
print("Хэш сообщения", hash_code_msg)
#подпись сообщения s=Sa(m) = m^d mod n
def signature_msg(hash_code,n,d):
    sign = (hash_code**d)%n
    return sign

sign_msg = signature_msg(hash_code_msg,n,d)
print("Значение подписи: {}".format(sign_msg))
#передаём пару m,s
def check_signature(sign_msg, n,e):
    check = (sign_msg**e) % n
    return check

check_sign = check_signature(sign_msg,n,e)
print("Значение проверки подписи = {}".format(check_sign))
```

Тестирование

Фраза по варианту

```
PS C:\Users\xiaomi\Desktop\cryptography_ciphers\lab_8\rsa> python3 main.py
Введите p: 31
True
Введите q: 7
True
N = 217
El = 180
E = 257
1 E подходит
D = 173
Открытый ключ e=257 n=217
Секретный ключ d=173 n=217
Введите сообщение:красивыми словами пастернак не помаслишь
Длина исходного сообщения 40 символов
Хэш сообщения 128
Значение подписи: 95
Значение проверки подписи = 128
```

Проверка текста на 1000 символов

```
PS C:\Users\xiaomi\Desktop\cryptography_ciphers\lab_8\rsa> python3 main.py
Введите p: 31
True
Введите q: 7
True
N = 217
El = 180
E = 257
1 E подходит
D = 173
Открытый ключ e=257 n=217
Секретный ключ d=173 n=217
Введите сообщение:вот пример статьи на тысячу символов. это достаточно маленький текст, оптимально подходящий для карточек товаров в интернет или магазинах или для н
ебольших информационных публикаций. в таком тексте редко бывает более двух или трех абзацев и обычно один подзаголовок. но можно и без него. на тысячу символов реком
ендовано использовать один или два ключа и одну картину. текст на тысячу символов это сколько примерно слов. сатистика показывает, что тысяча включает в себя стопять
десят или двести слов средней величины. но, если злоупотреблять предлогами, союзами и другими частями речи на один или два символа, то количество слов неизменно возр
астает. в копирайтерской деятельности принято считать тысячи с пробелами или без. учет пробелов увеличивает обьем текста примерно на сто или двести символов именно с
только раз мы разделяем слова свободным пространством. считать пробелы заказчики не любят, так как это пустое место. однако некоторые фирмы и биржи видят справедливы
м ставить стоимость за тысячу символов с пробелами, считая последние важным элементом качественного восприятия. согласитесь, читать слитный текст без единого пропуск
а, никто не будет. но большинству нужна цена за тысячу знаков без пробелов.
Длина исходного сообщения 1212 символов
Хэш сообщения 102
Значение подписи: 121
Значение проверки подписи = 102
PS C:\Users\xiaomi\Desktop\cryptography_ciphers\lab_8\rsa>
```