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 H: АСИММЕТРИЧНЫЕ ШИФРЫ

- **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

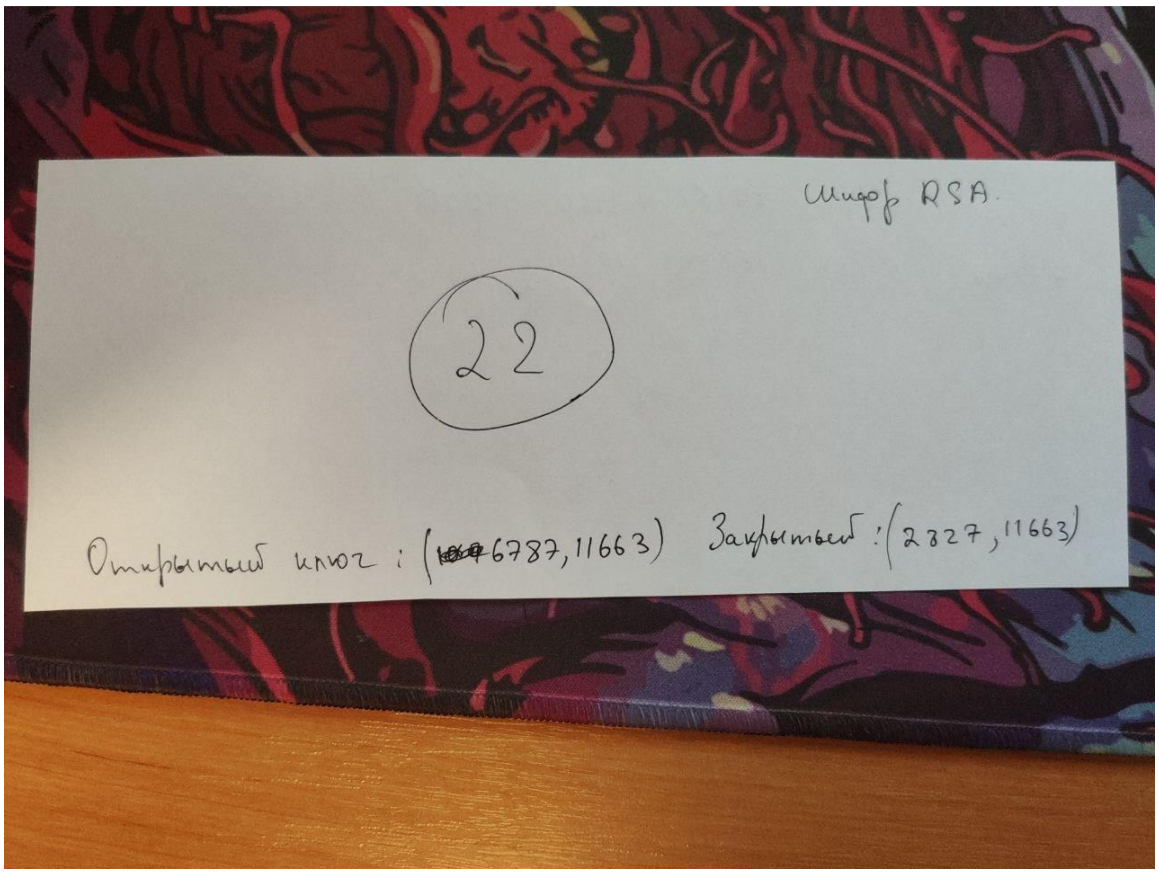
```
# -*- coding:utf-8 -*-

import random

def gcd(a, b):
    while b != 0:
        a, b = b, a % b
    return a

def multiplicative_inverse(e,r):
    for i in range(r):
        if((e*i)%r == 1):
            return i

def is_prime(num):
    if num == 2:
        return True
    if num < 2 or num % 2 == 0:
        return False
    for n in range(3, int(num**0.5)+2, 2):
        if num % n == 0:
            return False
    return True

def generate_keypair(p, q):
    if not (is_prime(p) and is_prime(q)):
        raise ValueError('Both numbers must be prime.')
    elif p == q:
        raise ValueError('p and q cannot be equal')
    #n = pq
    n = p * q

    phi = (p-1) * (q-1)

    e = random.randrange(1, phi)

    g = gcd(e, phi)
    while g != 1:
        e = random.randrange(1, phi)
        g = gcd(e, phi)
    d = multiplicative_inverse(e, phi)
    return ((e, n), (d, n))

def encrypt(pk, plaintext):
    key, n = pk
    cipher = [(ord(char) ** key) % n for char in plaintext]
```

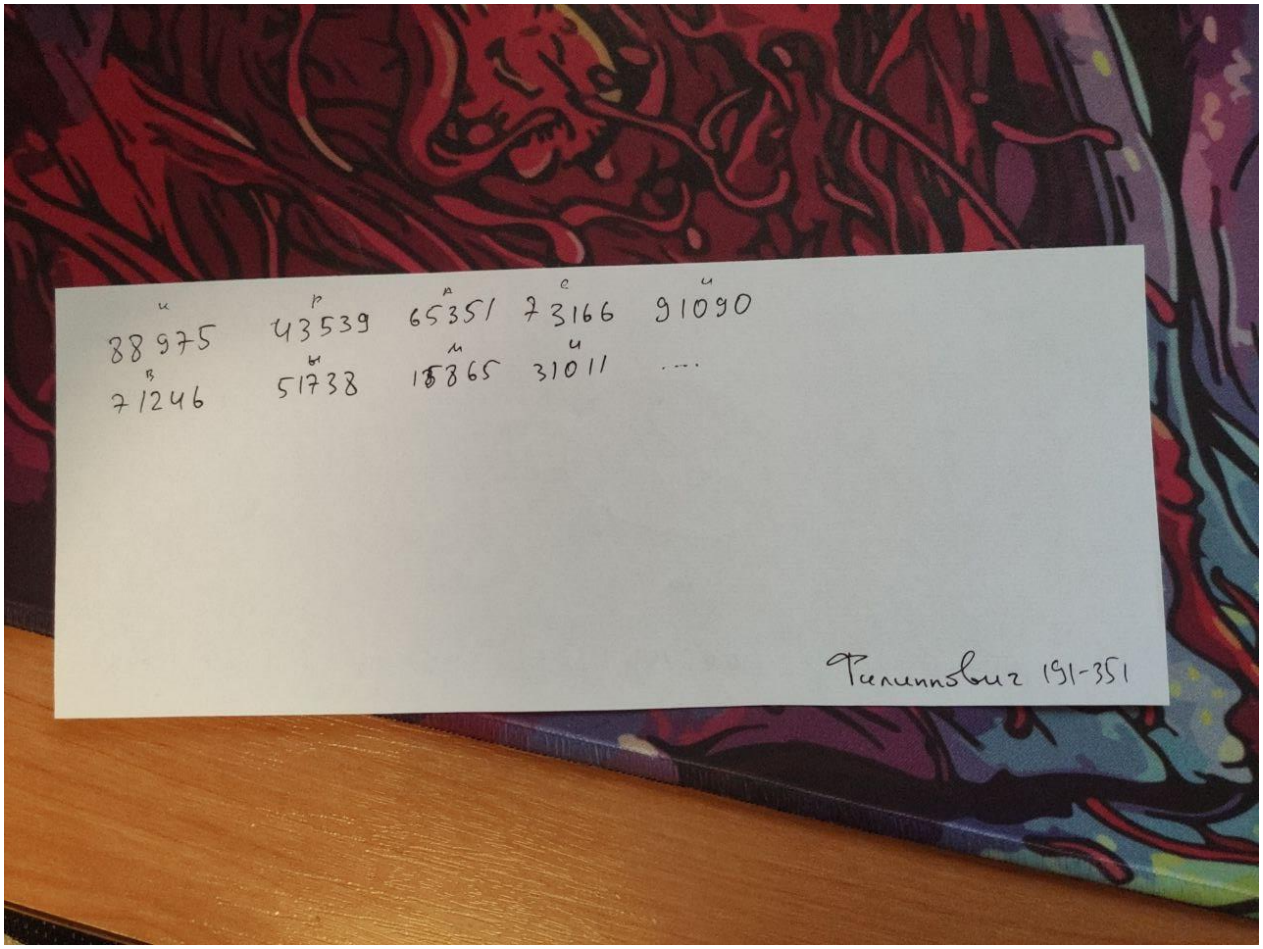
```python
    return cipher

def decrypt(pk, ciphertext):
    key, n = pk
    plain = [chr((char ** key) % n) for char in ciphertext]
    return ''.join(plain)


if __name__ == '__main__':
    '''
    Detect if the script is being run directly by the user
    '''
    print("RSA")
    p = int(input("Введите p: "))
    q = int(input("Введите q: "))
    public, private = generate_keypair(p, q)
    print("Публичный ключ: ", public ,"Секретный ключ: ", private)
    message = input("Введите сообщение: ")
    encrypted_msg = encrypt(private, message)
    print("Зашифрованное сообщение: ")
    print(''.join([str(x) for x in encrypted_msg]))
    print("Расшифрованное сообщение: ")
    print(decrypt(public, encrypted_msg))
```

Тестирование

Фраза по варианту

```
PS C:\Users\xiaomi\Desktop\cryptography_ciphers\lab_8\rsa> python3 .\rsa.py
RSA
Введите p: 107
Введите q: 109
Публичный ключ:  (6787, 11663) Секретный ключ:  (2827, 11663)
Введите сообщение: Красивыми словами пастернак не помаслишь
Зашифрованное сообщение:
33265997963638142085281818751106120855636381474248682818963611061208556362917963638145341211159971023996361970563610239211156362917868110619636381474242085452616 21
Расшифрованное сообщение:
Красивыми словами пастернак не помаслишь
```





Проверка текста на 1000 символов

```
PS C:\Users\xiaomi\Desktop\cryptography_ciphers\lab_8\rsa> python3 .\rsa.py
RSA
Введите p: 107
Введите q: 109
Публичный ключ:  (8165, 11663) Секретный ключ:  (9293, 11663)
Введите сообщение: 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 трех абзацев и обычно один подзаголовок. но можно и без него. на тысячу символов реко
мендовано использовать один или два ключа и одну картину. текст на тысячу символов это сколько примерно слов. сатистика показывает, что тысяча включает в себя стопят
ьдесят или двести слов средней величины. но, если злоупотреблять предлогами, союзами и другими частями речи на один или два символа, то количество слов неизменно воз
растает. 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 справедлив
ым ставить стоимость за тысячу символов с пробелами, считая последние важным элементом качественного восприятия. согласитесь, читать слитный текст без единого пропус
ка, никто не будет. но большинству нужна цена за тысячу знаков без пробелов.
Зашифрованное сообщение:
174887865341107728695781611180999150678161077238145341899453414189111801077221078994107725341104353814475836708610772381411180999117488786335887861748562310772147553
418786107729138878638145341899453418786836210787861077299918994335850621074189603611180111471077253415066036381453418816107728786869553411118099918994335841892107878
610772869587869138405887869138475107031118011147107729138335847510772603689947816534187868365066036107725341878617488994781687861748107721748107721118021075341506781
621075065341107721118033581118010772999189942862899480081118021078994405810772111803358111801077291383358475107722107506259387863358418955961118040581077211180210745
018786781699918994625611180878621072107104354058107728695708625933358111806036899462561118011147562310772174810772534189946036878699911077253415066036381453415061077
278165069138603687861077225931043517488994506534110772259387863358506506107729138174870864058107721118033581118010772534178165064058107728994259380088994625650617481
077211180107728786259310435836210787861077287869138111802107107728695878691388008899428628786335887861748878660365623107722107878610772999187865138210787861077211180
107722593506800810772210750628628786562310772210789941077253411043538144758367086107723814111809991174887863358878617481077278165066036878699915062107913887861748899
421078786107721118038148695878633584189800887861748899453414189107728786913811180210710772111803358111801077291381748899410772603633581055883689941077211180107728786
913821077086107726036899478165341111802107708656231077253415066036381453411077221078994107725341104353814475836708610772381411180999117488786335887861748107721475534
187861077238146036878633584189603687861077286957816111809991506781621078786107723814335887861748562310772381489945341111803814534111180603689941077286958786603689948
008104351748899450653418816107728365341878610772534110435381447583689941077217486036335810558836899450653411077217481077238145062593475107723814534187868695475534141
899138506381447553411077211180335811180107729138174850638145341111801077238143358878617481077238147816506913821075061114710772174850633581118083611180210710435562310
772210787868816107725063814335811180107728008335887867086869587865341781650625933358475534141891077286957816506913833588786286289949991111808816107723814878610558800
889949991111801077211180107729138781670862862111809991111801077283689943814534147599911118010772781650683611180107722107899410772878691381118021071077211180335811180
107729138174889941077238141118099911748878633588994881610772534187861077260368786335811180836506381453411748878610772381433588786174810772210750611180800899915062107
210787861077217488786800878168994381453418994506534156231077217481077260368786869511180781689941114753415067816381460368786111471077291385064755341506335841892107878
638145341111801077286957816111802107475534187861077238148361118053418994534141891077253411043538144758361118010772381410772869578168786259350633588994999111180107721
118033581118010772259350680085623107727086836506534110772869578168786259350633588786174810772708617485063358111808361118017488994506534110772878625939442506999110772
534150660363814534189941077286957816111809991506781621078786107722107899410772381453418786107721118033581118010772913817485063814534111180107723814111809991174887863
358878617481077211180999150621072107878610772381453418786335841896036878610772781689948008107729991104351077278168994800891385063358475506999110772381433588786174889
941077238141748878625938786913821071043599911077286957816878638145341781689942107381453411748878699915623107723814836111805341899453414189107728695781687862593506335
810435107728008899460368994800883611180603611180107722107506107723358105582593475534188161077253418994603610772603689946036107721475534187861077286957086381453418786
506107729991506381453418786562310772878691382107899460368786107722107506603687865341878678161043550610772450111180781699911043510772111801077225931118078165138111801
077217481118091384755341107723814869578168994174850691383358111801748104359991107723814534189941748111805341418910772381453418786111809991878638145341418910772800889
941077253411043538144758367086107723814111809991174887863358878617481077238141077286957816878625935063358899499911118088161077238148361118053418994475107728695878638
143358506913821071118050610772174889945138210710435999110772147533585069991506210753418786999110772603689948365063814534117485062107210787862862878610772174887863814
869578161118047553411118047556231077238148786286233588994381411180534150638144189881610772836111805341899453414189107723814335811180534121071043511147107725341506603
```

```
889949991111801077211180107729138781670862862111809991111801077283689943814534147599911118010772781650683611180107722107899410772878691381118021071077211180335811180
107729138174889941077238141118099911748878633588994881610772534187861077260368786335811180836506381453411748878610772381433588786174810772210750611180800899915062107
210787861077217488786800878168994381453418994506534156231077217481077260368786869511180781689941114753415067816381460368786111471077291385064755341506335841892107878
638145341111801077286957816111802107475534187861077238148361118053418994534141891077253411043538144758361118010772381410772869578168786259350633588994999111180107721
118033581118010772259350680085623107727086836506534110772869578168786259350633588786174810772708617485063358111808361118017488994506534110772878625939442506999110772
534150660363814534189941077286957816111809991506781621078786107722107899410772381453418786107721118033581118010772913817485063814534111180107723814111809991174887863
358878617481077211180999150621072107878610772381453418786335841896036878610772781689948008107729991104351077278168994800891385063358475506999110772381433588786174889
941077238141748878625938786913821071043599911077286957816878638145341781689942107381453411748878699915623107723814836111805341899453414189107728695781687862593506335
810435107728008899460368994800883611180603611180107722107506107723358105582593475534188161077253418994603610772603689946036107721475534187861077286957086381453418786
506107729991506381453418786562310772878691382107899460368786107722107506603687865341878678161043550610772450111180781699911043510772111801077225931118078165138111801
077217481118091384755341107723814869578168994174850691383358111801748104359991107723814534189941748111805341418910772381453418786111809991878638145341418910772800889
941077253411043538144758367086107723814111809991174887863358878617481077238141077286957816878625935063358899499911118088161077238148361118053418994475107728695878638
143358506913821071118050610772174889945138210710435999110772147533585069991506210753418786999110772603689948365063814534117485062107210787862862878610772174887863814
869578161118047553411118047556231077238148786286233588994381411180534150638144189881610772836111805341899453414189107723814335811180534121071043511147107725341506603
638145341107722593506800810772506913811180210787862862878610772869578168786869570863814603689948816107722107111806036534187861077221075061077225937086913850653415623
107722107878610772259387863358418955961118021073814534117487086107722107708651382107899410772625650621078994107728008899410772534110435381447583670861077280082107899
460368786174810772259350680081077286957816878625935063358878617485623
Расшифрованное сообщение:
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 трех 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. сатистика показывает, что тысяча включает в себя стопятьдесят или двести с
лов средней величины. но, если злоупотреблять предлогами, союзами и другими частями речи на один или два символа, то количество слов неизменно возрастает. в копирайт
ерской деятельности принято считать тысячи с пробелами или без. учет пробелов увеличивает объем текста примерно на сто или двести символов именно столько раз мы разд
еляем слова свободным пространством. считать пробелы заказчики не любят, так как это пустое место. однако некоторые фирмы и биржи видят справедливым ставить стоимост
ь 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.
PS C:\Users\xiaomi\Desktop\cryptography_ciphers\lab_8\rsa>
```