一、 9.3题，已知密文和n和公钥e，求明文。

C=10,e=5,n=35=5\*7

φ(n)=4\*6=24

e\*d=1(mod φ(n)),即5d=1(mod 24),d=5

明文M=Cd=105=5(mod n)

二、 9.4题，已知公钥e和n，求私钥。

e=31,n=3599.

首先对n进行分解，3599=59×61。

φ(n)=58\*60=3480

由ed%φ(n)=1,求31关于3480的逆元，用扩展欧几里得算法：

3480%31=8 8=(1,-112)

31%8=7 7=(0,1)-3\*(1,-112)=(-3,337)

8%7=1 1=(1,-112)-1\*(-3,337)=(4,-449)

即：1=4\*3480-31\*449

也就是31关于3480的逆元为-449，也就是3031

三、9.7题，在RSA公钥加密体制中，每一个用户都有一个公钥和一个私钥。假设Bob泄露了他的私钥，为了偷懒，他决定在不改变大整数n的情况下，重新产生一对公钥和私钥。这样安全吗？

不安全。设公钥为e，私钥为d，则ed%φ(n)=1

因为Bob泄露了私钥，所以人们可以用大整数乘法轻易求出ed来，设k=ed-1，则k为φ(n)的整数倍。

由欧拉定理,对于任意与n互质的数字a，也就是[2,n-1]区间上除了p和q，都有：

所以

下面的过程类似Miller-Labin测试：将k分解为k=2tx，在[2,n-1]上任取整数a，对于序列

设y是如上序列中的一个数字，那么y的前一个数字是sqrt(y)。如果y模n等于1，而sqrt(y)模n不等于1，求gcd(sqrt(y)-1,n)就能得到n的一个因子。

*"""  
在RSA中，给一对公钥和私钥，就可以完成n的大整数分解  
"""*p, q = 7, 13  
n = p \* q  
phi = (p - 1) \* (q - 1)  
k = phi \* 9*#这里直接假设k=ed-1=9\*phi***def** gcd(x, y):  
 **return** x **if** y == 0 **else** gcd(y, x % y)  
  
  
**def** pow(a, x, n):  
 *#模幂乘法* ans = 1  
 **for** i **in** range(x):  
 ans = (ans \* a) % n  
 **return** ans  
  
  
**def** test(a):  
 x = k  
 **while** pow(a, x, n) % n == 1 **and** x % 2 == 0: x //= 2  
 print(gcd(pow(a, x, n) - 1, n), a, x, pow(a, x, n))  
  
  
**import** random  
  
**for** i **in** range(2, n):  
 test(random.randint(2, n))

输出结果如下：

第一列表示求得的，第二列表示随机选择的数字a，第三个数字x表示幂，第四个数字表示，可以看到7，9大量出现。

7 34 162 64

7 15 162 64

13 55 81 27

7 57 162 64

7 33 162 64

7 13 648 78

7 57 162 64

13 49 648 14

91 9 81 1

7 60 162 64

7 67 162 64

7 24 162 64

7 59 162 64

7 64 81 64

13 56 648 14

7 78 648 78

7 30 81 64

13 35 648 14

91 29 81 1

7 8 162 64

7 73 162 64

13 14 648 14

13 87 81 27

91 16 81 1

7 89 162 64

7 36 81 64

1 62 81 90

1 90 81 90

13 42 648 14

7 45 162 64

7 26 648 78

1 90 81 90

7 89 162 64

7 19 162 64

7 52 648 78

7 73 162 64

7 67 162 64

7 41 162 64

13 27 81 27

1 90 81 90

7 51 81 64

7 20 162 64

7 78 648 78

13 55 81 27

7 58 162 64

1 62 81 90

7 54 162 64

7 5 162 64

7 67 162 64

13 56 648 14

7 32 162 64

13 3 81 27

13 42 648 14

7 33 162 64

7 57 162 64

13 7 648 14

13 66 81 27

7 86 162 64

13 21 648 14

7 47 162 64

7 58 162 64

7 43 81 64

7 5 162 64

7 73 162 64

1 17 81 90

7 80 162 64

13 68 81 27

91 81 81 1

13 21 648 14

1 38 81 90

7 4 81 64

7 67 162 64

7 86 162 64

1 75 81 90

13 35 648 14

13 3 81 27

13 42 648 14

13 42 648 14

7 25 81 64

7 85 162 64

13 7 648 14

7 11 162 64

13 77 648 14

7 26 648 78

7 85 162 64

13 7 648 14

7 2 162 64

7 51 81 64

13 61 81 27