1. Implement Extended Euclid algorithm to find GCD and Multiplicative inverse.

Problem -1 : 18 $^{-1}$ mod 557; x=18 m=557

A1	A2	A3	B1	B2	B3	T1	T2	T3	Q
1	0	557	0	1	18	1	-30	17	30
0	1	18	1	-30	17	-1	31	1	1
1	-30	17	-1	31	1				

B3=1 so value of B2 is the inverse value. 18 $^{-1}$ mod 557 =31

Output :

```
C:\Users\hp\Desktop\College\Sem 7\INS\Practicals\2_1>C:/Users/hp/AppData/Local/Programs/Pyt
hon/Python38/python.exe "c:/Users/hp/Desktop/College/Sem 7/INS/Practicals/2_1/P2_1.py"
x^-1 mod m
Enter x : 18
Enter m : 557
GCD :  1
Multiplicative Inverse :  31

C:\Users\hp\Desktop\College\Sem 7\INS\Practicals\2_1>
```

Problem – 2: Multiplicative inverse of 37 mod 1023

A1	A2	A3	B1	B2	B3	T1	T2	T3	Q
1	0	1023	0	1	37	1	-27	24	27
0	1	37	1	-27	24	-1	28	13	1
1	-27	24	-1	28	13	2	-55	11	1
-1	28	13	2	-55	11	-3	83	2	1
2	-55	11	-3	83	2	5	-470	1	5
-3	83	2	5	-470	1				

-470 mod 1023 = 1023*-1 + 553 = 553

Program :

```
def mulInv(x,m):
    a1,a2,a3 = 1,0,m
    b1,b2,b3 = 0,1,x
    while True:
        if b3==0:
            return a3, 'Not Exist'
        if b3==1:
            return 1,b2
        q = a3//b3
        a1,a2,a3 , b1,b2,b3 = b1,b2,b3 , a1-q*b1, a2-q*b2, a3-q*b3

print("x^-1 mod m")
```

```
x = int(input("Enter x : "))
m = int(input("Enter m : "))

gcd, inv = mulInv(x,m)
if inv<0:
   inv = m+inv

print('GCD : ',gcd)
print('Multiplicative Inverse : ', inv)
```

Output :

```
C:\Users\hp\Desktop\College\Sem 7\INS\Practicals\2_1>C:/Users/hp/AppData/Local/Programs/Python/Python38/python.exe "c:/Users/hp/Desktop/College/Sem 7/INS/Practicals/2_1/P2_1.py"
x^-1 mod m
Enter x : 37
Enter m : 1023
GCD :  1
Multiplicative Inverse :  553
```

2. Implement RSA algorithm.

- Take two prime numbers p,q
  n=pxq
- Initially take encryption key such that it is relatively prime with ϕ(n).
- Find out decryption key.
- Take plaintext message M, Ciphertext C=Me mod n.
- To get plaintext from ciphertext M=Cd mod n.
- Test case :

  Two prime numbers 17,11

  Encryption key = 7

  Decryption key = 23

  M=88

  C=11

To find decryption key, apply extended Euclidean algorithm.

Program :

```
import math
def mulInv(x,m):
   a1,a2,a3 = 1,0,m
   b1,b2,b3 = 0,1,x
   while True:
      if b3==0:
         return a3, 'Not Exist'
      if b3==1:
```

```
            return 1,b2
        q = a3//b3
        a1,a2,a3 , b1,b2,b3 = b1,b2,b3 , a1-q*b1, a2-q*b2, a3-q*b3

def rsa(p,q):
    n = p*q
    fi = (p-1)*(q-1)

    for e in range(2,fi):
        if math.gcd(e,fi)== 1:
            break
    gcd, d = mulInv(e,fi)

    if d<0:
        d = fi+d
    return e,d,n

def encrypt(M,e,n):
    return M**e % n

def decrypt(C,d,n):
    return C**d % n

p = int(input('Enter the value of p = '))
q = int(input('Enter the value of q = '))
M = int(input('Enter Message : '))

e,d,n = rsa(p,q)
C = encrypt(M,e,n)
M = decrypt(C,d,n)

print('Encryption Key : ',(e,n))
print('Decryption Key : ',(d,n))
print('Cipher Text : ',C)
print('Message  : ',M)
```

Output :

```
C:\Users\hp\Desktop\College\Sem 7\INS\Practicals\2_1>C:/Users/hp/AppData/Local/Programs/
Python/Python38/python.exe "c:/Users/hp/Desktop/College/Sem 7/INS/Practicals/2_1/P2_1.py
"
Enter the value of p = 17
Enter the value of q = 11
Enter Message : 88
Encryption Key :  (3, 187)
Decryption Key :  (107, 187)
Cipher Text :  44
Message  :  88
```