CODE :

```
try:
   input = raw_input
except NameError:
pass try:

   chr = unichr
except NameError:
pass

p=int(input('Enter prime p: '))
q=int(input('Enter prime q: '))
print("Choosen primes:\np=" + str(p) + ", q=" + str(q) + "\n")
n=p*q

print("n = p * q = " + str(n) + "\n")
phi=(p-1)*(q-1)
print("Euler's function (totient) [phi(n)]: " + str(phi) +
"\n")
def gcd(a, b):
while b != 0: c=a%b

a=b

b=c return a

def modinv(a, m):
    for x in range(1, m):
        if (a * x) % m == 1:
            return x
    return None
def coprimes(a):
    l = []
    for x in range(2, a):
        if gcd(a, x) == 1 and modinv(x,phi) != None and x <
(p-1) and x < (q-1):
            l.append(x)
    for x in l:
        if x == modinv(x,phi):
            l.remove(x)
    return l
def encrypt(p,k, plaintext):
    #Unpack the key into it's components
key, n = p,k

#Convert each letter in the plaintext to numbers based on the character using a^b mod m

    cipher = [(ord(char) ** key) % n for char in plaintext]
    #Return the array of bytes
    return cipher
def decrypt(p,k, ciphertext):
    #Unpack the key into its components
    key, n = p,k

#Generate the plaintext based on the ciphertext and key
using a^b mod m
    plain = [chr((char ** key) % n) for char in ciphertext]
    #Return the array of bytes as a string
    return ''.join(plain)
print("Choose an e from a below coprimes array:\n")
print(str(coprimes(phi)) + "\n")
e=int(input())
d=modinv(e,phi)
print("\nYour public key is a pair of numbers (e=" + str(e) +
", n=" + str(n) + ").\n")
print("Your private key is a pair of numbers (d=" + str(d) +
", n=" + str(n) + ").\n")
plaintext=input("Enter plaintext : ")
print ("\n\nEncrypting message with public key (", d,",",n ,")
. . .")
print ("\nYour ciphertext is:")
emsg = encrypt(d,n,plaintext)
print ("\t\t",''.join(map(lambda x: str(x), emsg)))
print ("\n\nDecrypting message with public key (", e,",",n ,")
. . .")
print ("\nYour message is:")
print ("\t\t",decrypt(e,n, emsg))
```

```
OUTPUT :
Enter prime p: 17
Enter prime q: 11
Chosen primes:
p=17, q=11

n = p * q = 187
Euler's function (totient) [phi(n)]: 160
Choose an e from a below coprimes array:
[3, 7, 9]
9
Your public key is a pair of numbers (e=9, n=187). Your private key is a pair of numbers (d=89,
n=187). Enter plaintext : hello, how are you?

Encrypting message with public key ( 89 , 187 ) . . .
Your ciphertext is:
53501811811111433253111170325124503212111115173
Decrypting message with public key (9, 187) . . .
Your message is:
            hello, how are you?
```