In [1]:

```
def gcd(a, b): # calculates GCD of a and d
    while b != 0:
        c = a % b
        a = b
        b = c
    return a


def modinv(a, m): # calculates modulo inverse of a for mod m
    for x in range(1, m):
        if (a * x) % m == 1:
            return x
    return None


def coprimes(a): # calculates all possible co-prime numbers with a
    l = []
    for x in range(2, a):
        if gcd(a, x) == 1 and modinv(x, phi) != None:
            l.append(x)
    for x in l:
        if x == modinv(x, phi):
            l.remove(x)
    return l

def encrypt_block(m): # encrypts a single block
    c = m ** e % n
    return c


def decrypt_block(c): # decrypts a single block
    m = c ** d % n
    return m


def encrypt_string(s): # applies encryption
    return ''.join([chr(encrypt_block(ord(x))) for x in list(s)])


def decrypt_string(s): # applies decryption
    return ''.join([chr(decrypt_block(ord(x))) for x in list(s)])

if __name__ == "__main__":
    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")

    print("Choose an e from a below coprimes array:\n")
    print(str(coprimes(phi)) + "\n")
    e = int(input())

    d = modinv(e, phi) # calculates the decryption key d

    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
")

    s = input("Enter a message to encrypt: ")
    print("\nPlain message: " + s + "\n")
```

```
    enc = encrypt_string(s)
    print("Encrypted message: ", enc, "\n")
    dec = decrypt_string(enc)
    print("Decrypted message: " + dec + "\n")
```

```
Enter prime p: 17
Enter prime q: 13
Choosen primes:
p=17, q=13

n = p * q = 221

Euler's function (totient) [phi(n)]: 192

Choose an e from a below coprimes array:

[5, 7, 11, 13, 17, 19, 23, 25, 29, 35, 37, 41, 43, 47, 49, 53, 55, 59, 61, 67, 71, 73, 77,
79, 83, 85, 89, 91, 97, 101, 103, 107, 109, 113, 115, 119, 121, 125, 131, 133, 137, 139,
143, 145, 149, 151, 155, 157, 163, 167, 169, 173, 175, 179, 181, 185, 187]

7

Your public key is a pair of numbers (e=7, n=221).

Your private key is a pair of numbers (d=55, n=221).

Enter a message to encrypt: Hello I am Jessica

Plain message: Hello I am Jessica

Encrypted message:  eRR;,;Ø;0ejjO9

Decrypted message: Hello I am Jessica


In [ ]:
```