def gcd(a, b):

if a == 0:

return b

return gcd(b % a, a)

P = 53

Q = 59

n = P \* Q

phi\_n = (P-1) \* (Q-1)

# Generating e

e = 2

while e < phi\_n:

if gcd(e, phi\_n) == 1:

break

e += 1

# Generating d

k = 1

while (k \* phi\_n + 1) % e != 0:

k += 1

d = (k \* phi\_n + 1) // e # // used for floor division

U = [e, n] # Public key

R = [d, n] # Private key

print("Primes:\t\t", P, ",", Q)

print("N:\t\t", n)

print("phi(N):\t\t", phi\_n)

print("e:\t\t", e)

print("d:\t\t", d)

print("Public key:\t", "[e, n] =", U)

print("Private key:\t", "[d, n] =", R)

plaintext = int(input("Enter Plaintext"))

C= (plaintext\*\*e) % n

print(C)

P= (C\*\*d) % n

print(P)

if P== plaintext:

print("Message is succesfully shared")

else:

print("Not shared")