Rsa

```
import math



def gcd(a, h):
temp = 0
while(1):
temp = a % h
if (temp == 0):
return h
a = h
h = temp



p = 3
q = 7
n = p*q
e = 2
phi = (p-1)*(q-1)


while (e < phi):


# e must be co-prime to phi and
# smaller than phi.
if(gcd(e, phi) == 1):
break
else:
e = e+1


# Private key (d stands for decrypt)
# choosing d such that it satisfies
# d*e = 1 + k * totient
```

```
k = 2
d = (1 + (k*phi))/e


# Message to be encrypted
msg = 13


print("Message data = ", msg)


# Encryption c = (msg ^ e) % n
c = pow(msg, e)
c = math.fmod(c, n)
print("Encrypted data = ", c)


# Decryption m = (c ^ d) % n
m = pow(c, d)
m = math.fmod(m, n)
print("Original Message Sent = ", m)
```