**RSA Algorithm**

def gcd(a, b):  
if b == 0:  
return a  
return gcd(b, a % b)  
def mod\_inverse(a, m):  
a = a % m  
for x in range(1, m):  
if (a \* x) % m == 1:  
return x  
return -1  
def power(x, y, p):  
result = 1  
x = x % p  
while y > 0:  
if y & 1:  
result = (result \* x) % p  
y = y >> 1  
x = (x \* x) % p  
return result  
def generate\_keys(p, q):  
n = p \* q  
phi = (p - 1) \* (q - 1)  
e = 3  
while gcd(e, phi) != 1:  
e += 1  
d = mod\_inverse(e, phi)  
return e, d, n

def encrypt(plaintext, e, n):  
return power(plaintext, e, n)

def decrypt(ciphertext, d, n):  
return power(ciphertext, d, n)

