Course Name: Cryptography and Network Security

Course Code: BCSE309P

Lab 10: ELGAMMAL ENCRYPTION AND DECRYPTION

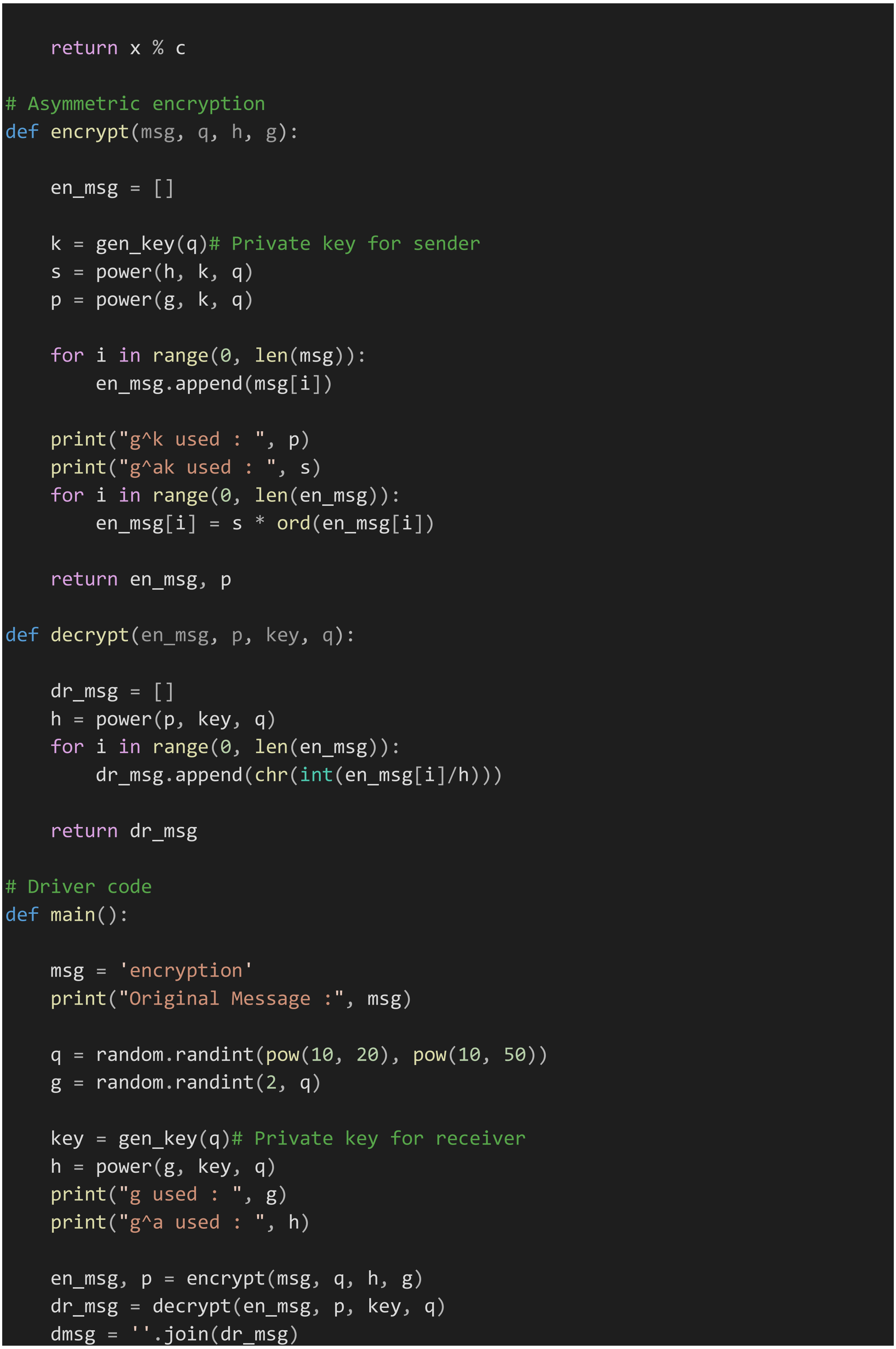
Name: Shlok Kamath

Registration Number: 21BAI1844

Professor: Dr. Rajesh R

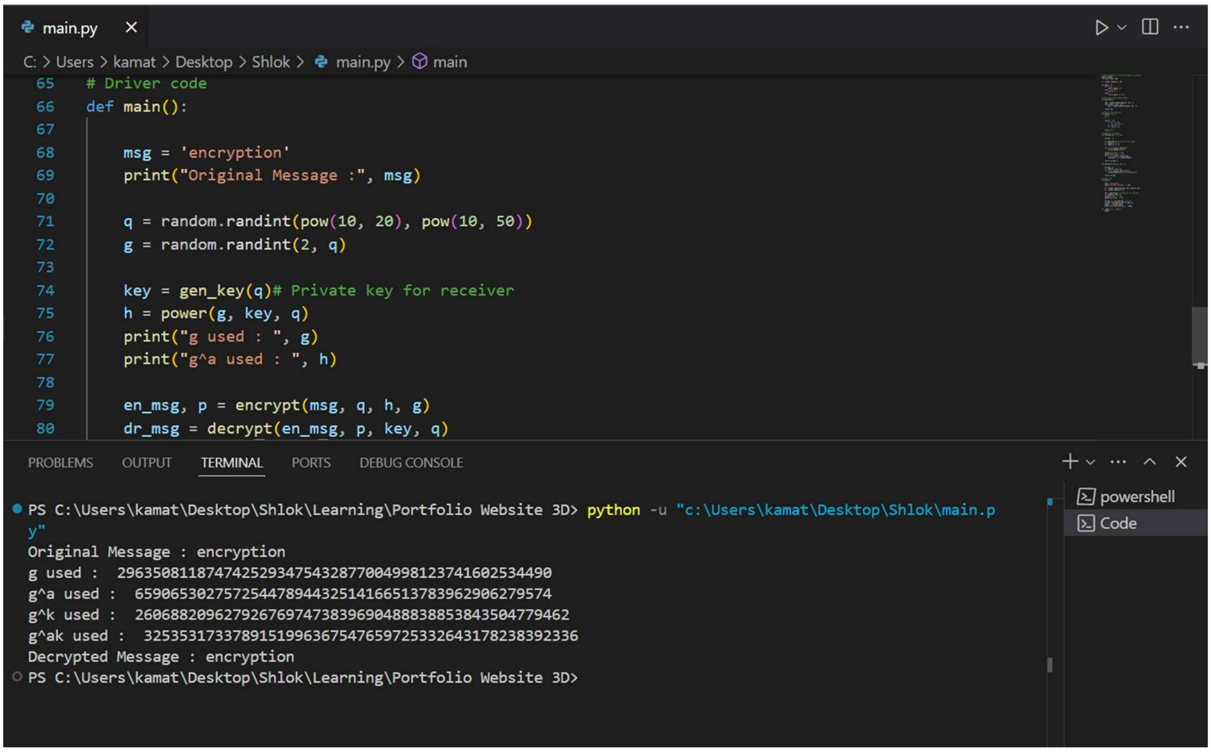
Code:

```
# Python program to illustrate ElGamal encryption
import random
from math import pow

a = random.randint(2, 10)

def gcd(a, b):
    if a < b:
        return gcd(b, a)
    elif a % b == 0:
        return b;
    else:
        return gcd(b, a % b)

# Generating large random numbers
def gen_key(q):

    key = random.randint(pow(10, 20), q)
    while gcd(q, key) != 1:
        key = random.randint(pow(10, 20), q)

    return key

# Modular exponentiation
def power(a, b, c):
    x = 1
    y = a

    while b > 0:
        if b % 2 != 0:
            x = (x * y) % c;
        y = (y * y) % c
        b = int(b / 2)
```



```
    return x % c

# Asymmetric encryption
def encrypt(msg, q, h, g):

    en_msg = []

    k = gen_key(q)# Private key for sender
    s = power(h, k, q)
    p = power(g, k, q)

    for i in range(0, len(msg)):
        en_msg.append(msg[i])

    print("g^k used : ", p)
    print("g^ak used : ", s)
    for i in range(0, len(en_msg)):
        en_msg[i] = s * ord(en_msg[i])

    return en_msg, p

def decrypt(en_msg, p, key, q):

    dr_msg = []
    h = power(p, key, q)
    for i in range(0, len(en_msg)):
        dr_msg.append(chr(int(en_msg[i]/h)))

    return dr_msg

# Driver code
def main():

    msg = 'encryption'
    print("Original Message :", msg)

    q = random.randint(pow(10, 20), pow(10, 50))
    g = random.randint(2, q)

    key = gen_key(q)# Private key for receiver
    h = power(g, key, q)
    print("g used : ", g)
    print("g^a used : ", h)

    en_msg, p = encrypt(msg, q, h, g)
    dr_msg = decrypt(en_msg, p, key, q)
    dmsg = ''.join(dr_msg)
    print("Decrypted Message :", dmsg);

if __name__ == '__main__':
    main()
```

Output:



```
# Driver code
def main():

    msg = 'encryption'
    print("Original Message :", msg)

    q = random.randint(pow(10, 20), pow(10, 50))
    g = random.randint(2, q)

    key = gen_key(q)# Private key for receiver
    h = power(g, key, q)
    print("g used : ", g)
    print("g^a used : ", h)

    en_msg, p = encrypt(msg, q, h, g)
    dr_msg = decrypt(en_msg, p, key, q)
```

```
PS C:\Users\kamat\Desktop\Shlok\Learning\Portfolio Website 3D> python -u "c:\Users\kamat\Desktop\Shlok\main.p
y"
Original Message : encryption
g used :  2963508118747425293475432877004998123741602534490
g^a used :  65906530275725447894432514166513783962906279574
g^k used :  2606882096279267697473839690488838853843504779462
g^ak used :  3253531733789151996367547659725332643178238392336
Decrypted Message : encryption
PS C:\Users\kamat\Desktop\Shlok\Learning\Portfolio Website 3D>
```