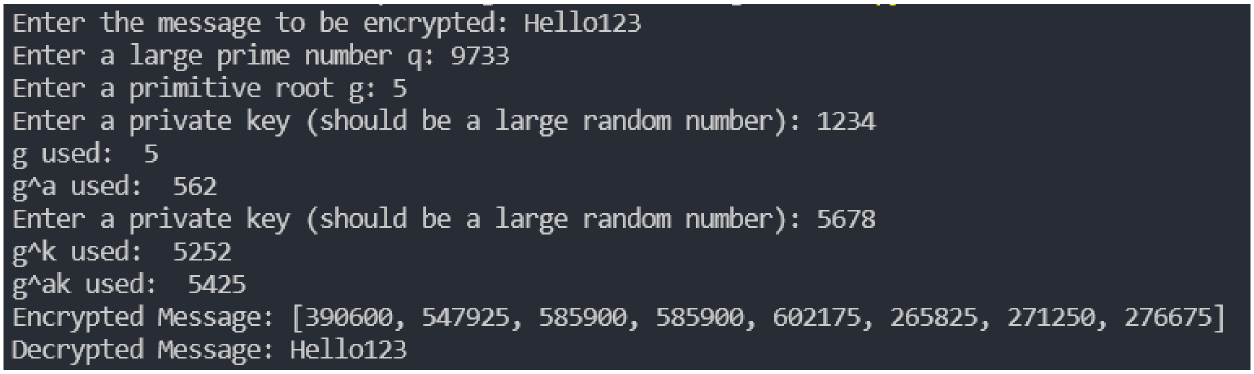
Assignment 7

ElGamal signature

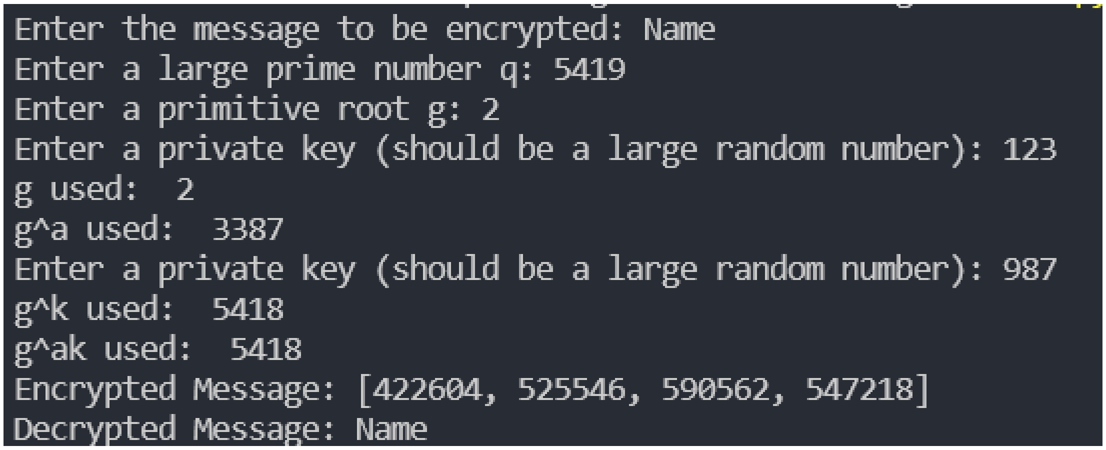
Code:-

```
from math import pow

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

def gen_key(q):
    key = int(input("Enter a private key (should be a large random number): "))
    while gcd(q, key) != 1:
        key = int(input("Private key should be co-prime with q. Enter another private key:
"))
    return key

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

def encrypt(msg, q, h, g):
    en_msg = []
    k = gen_key(q)
    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

def main():
    msg = input("Enter the message to be encrypted: ")
    q = int(input("Enter a large prime number q: "))
    g = int(input("Enter a primitive root g: "))
    key = gen_key(q)
    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("Encrypted Message:", en_msg)
    print("Decrypted Message:", dmsg)

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

Output:-



```
Enter the message to be encrypted: Hello123
Enter a large prime number q: 9733
Enter a primitive root g: 5
Enter a private key (should be a large random number): 1234
g used:  5
g^a used:  562
Enter a private key (should be a large random number): 5678
g^k used:  5252
g^ak used:  5425
Encrypted Message: [390600, 547925, 585900, 585900, 602175, 265825, 271250, 276675]
Decrypted Message: Hello123
```



```
Enter the message to be encrypted: Name
Enter a large prime number q: 5419
Enter a primitive root g: 2
Enter a private key (should be a large random number): 123
g used:  2
g^a used:  3387
Enter a private key (should be a large random number): 987
g^k used:  5418
g^ak used:  5418
Encrypted Message: [422604, 525546, 590562, 547218]
Decrypted Message: Name
```

Conclusion:-

ElGamal encryption is a secure method for encryption and digital signatures, utilizing discrete logarithm problem difficulty. Ensuring confidentiality and integrity requires large prime numbers and primitive roots.