**PRACTICAL 8**

**AIM**: Implement Diffi-Hellmen Key exchange Method.

**Code**:

q = int(input("enter the prime number for q: "))

alpha = int(input("enter the value of alpha: "))

a1 = int(input("enter the value for a1: "))

a2 = int(input("enter the value for a2: "))

y1 = alpha \*\* a1 % q

y2 = alpha \*\* a2 % q

print("y1: ", y1)

print("y2: ", y2)

#checking

k1 = y2 \*\* a1 % q

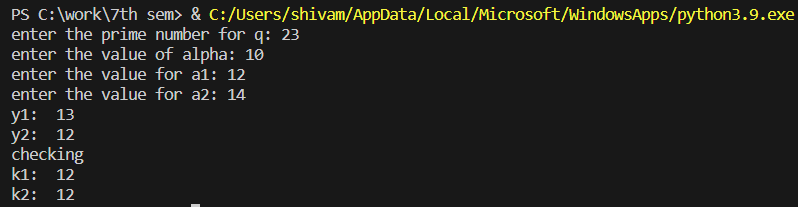
k2 = y1 \*\* a2 % q

print("checking")

print("k1: ", k1)

print("k2: ", k2)

output:



**PRACTICAL 9**

**AIM**: Implement RSA encryption & decryption algorithm.

Code:

import math

def gcd(a, h):

    temp = 0

    while(1):

        temp = a % h

        if (temp == 0):

            return h

        a = h

        h = temp

p = int(input("enter the value of p: "))

q = int(input("enter the value of q: "))

n = p\*q

e = int(input("enter the value of e: "))

phi = (p-1)\*(q-1)

while (e < phi):

    if(gcd(e, phi) == 1):

        break

    else:

        e = e+1

k = int(input("enter the value of k: "))

d = (1 + (k\*phi))/e

msg = 12.0

print("Message data = ", msg)

c = pow(msg, e)

c = math.fmod(c, n)

print("Encrypted data = ", c)

m = pow(c, d)

m = math.fmod(m, n)

print("Original Message Sent = ", m)

output:

