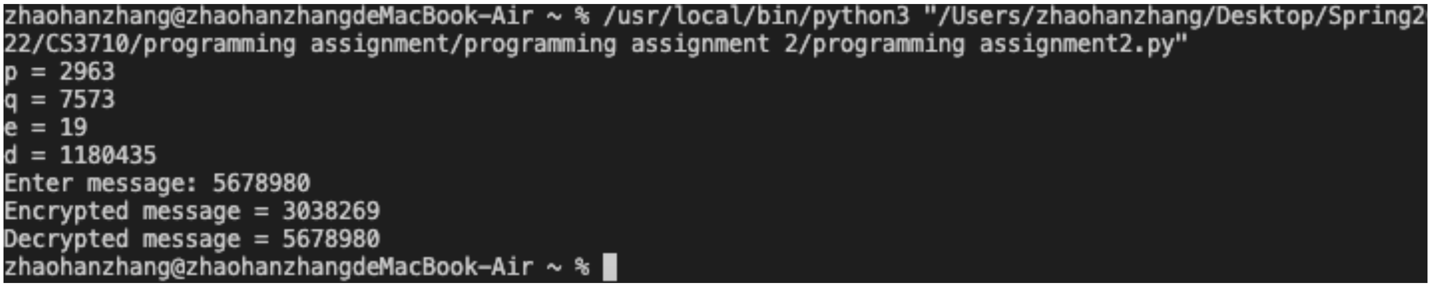
Source code:

```
import math
import random

# check if a number is prime
def check_prime(a):
    if(a==2):
        return True
    elif((a<2) or ((a%2)==0)):
        return False
    elif(a>2):
        for i in range(2,a):
            if not(a%i):
                return False
    return True

# set random p and q (there is performance issue so I set the limit to 10000)
p = int(random.random()*10000)
q = int(random.random()*10000)
while not(check_prime(p)):
    p = int(random.random()*10000)
while not(check_prime(q)):
    q = int(random.random()*10000)
# print out p and q
print("p = " + str(p))
print("q = " + str(q))

# calculate n
n = p*q

# calculate phi(n)
phi_n = (p-1)*(q-1)

# calculate e
# we have to make sure that e and phi_n are relative prime
# check if e and phi_n have a GCD of 1
def check_gcd(e,phi_n):
    while(phi_n!=0):
        e,phi_n=phi_n,e%phi_n
    return e
```

```
# find e and make it small for easier calculation
for i in range(20):
    if check_gcd(i, phi_n)==1:
        e = i
# print e out
print("e = " + str(e))

# calculate d
for i in range(phi_n):
    if (e*i)%phi_n == 1:
        d = i
        break
# print d
print("d = " + str(d))

# input message
M = int(input("Enter message: "))

# calculate ciphertext C
C = pow(M, e, n)
# print ciphertext C
print("Encrypted message = " + str(C))

# calculate original message M
ori_M = pow(C, d, n)
# print original message
print("Decrypted message = " + str(ori_M))
```

RESULT



```
zhaohanzhang@zhaohanzhangdeMacBook-Air ~ % /usr/local/bin/python3 "/Users/zhaohanzhang/Desktop/Spring2
22/CS3710/programming assignment/programming assignment 2/programming assignment2.py"
p = 2963
q = 7573
e = 19
d = 1180435
Enter message: 5678980
Encrypted message = 3038269
Decrypted message = 5678980
zhaohanzhang@zhaohanzhangdeMacBook-Air ~ % 
```