QUESTION 14:

import random

import math

prime = set()

public\_key = None

private\_key = None

n = None

def primefiller():

seive = [True] \* 250

seive[0] = False

seive[1] = False

for i in range(2, 250):

for j in range(i \* 2, 250, i):

seive[j] = False

for i in range(len(seive)):

if seive[i]:

prime.add(i)

def pickrandomprime():

global prime

k = random.randint(0, len(prime) - 1)

it = iter(prime)

for \_ in range(k):

next(it)

ret = next(it)

prime.remove(ret)

return ret

def setkeys():

global public\_key, private\_key, n

prime1 = pickrandomprime()

prime2 = pickrandomprime()

n = prime1 \* prime2

fi = (prime1 - 1) \* (prime2 - 1)

e = 2

while True:

if math.gcd(e, fi) == 1:

break

e += 1

public\_key = e

d = 2

while True:

if (d \* e) % fi == 1:

break

d += 1

private\_key = d

def encrypt(message):

global public\_key, n

e = public\_key

encrypted\_text = 1

while e > 0:

encrypted\_text \*= message

encrypted\_text %= n

e -= 1

return encrypted\_text

def decrypt(encrypted\_text):

global private\_key, n

d = private\_key

decrypted = 1

while d > 0:

decrypted \*= encrypted\_text

decrypted %= n

d -= 1

return decrypted

def encoder(message):

encoded = []

for letter in message:

encoded.append(encrypt(ord(letter)))

return encoded

def decoder(encoded):

s = ''

for num in encoded:

s += chr(decrypt(num))

return s

if \_\_name\_\_ == '\_\_main\_\_':

primefiller()

setkeys()

message = "Test Message"

coded = encoder(message)

print("Initial message:")

print(message)

print("\n\nThe encoded message(encrypted by public key)\n")

print(''.join(str(p) for p in coded))

print("\n\nThe decoded message(decrypted by public key)\n")

print(''.join(str(p) for p in decoder(coded)))

OUTPUT:

Initial message:

Test Message

The encoded message(encrypted by public key)

7746161612204304157781322161612201220561495791616

The decoded message(decrypted by public key)

Test Message