INS practical: RSA :

Code:

import random

def is\_prime(num):

    if num <= 1:

        return False

    if num <= 3:

        return True

    if num % 2 == 0 or num % 3 == 0:

        return False

    i = 5

    while i \* i <= num:

        if num % i == 0 or num % (i + 2) == 0:

            return False

        i += 6

    return True

def gcd(a, b):

    while b != 0:

        a, b = b, a % b

    return a

def modinv(a, m):

    m0, x0, x1 = m, 0, 1

    while a > 1:

        q = a // m

        m, a = a % m, m

        x0, x1 = x1 - q \* x0, x0

    return x1 + m0 if x1 < 0 else x1

def generate\_keypair(p, q):

    if not (is\_prime(p) and is\_prime(q)):

        raise ValueError("Both numbers must be prime.")

    if p == q:

        raise ValueError("p and q should not be equal.")

    n = p \* q

    phi = (p - 1) \* (q - 1)

    e = random.randrange(2, phi)

    while gcd(e, phi) != 1:

        e = random.randrange(2, phi)

    d = modinv(e, phi)

    return (e, n), (d, n)

def encrypt(public\_key, plaintext):

    e, n = public\_key

    return [pow(ord(char), e, n) for char in plaintext]

def decrypt(private\_key, ciphertext):

    d, n = private\_key

    return ''.join(chr(pow(char, d, n)) for char in ciphertext)

def main():

    try:

        p = int(input("Enter a prime number p: "))

        q = int(input("Enter a different prime number q: "))

    except ValueError:

        print("Invalid input. Please enter valid integers.")

        return

    try:

        public\_key, private\_key = generate\_keypair(p, q)

        print(f"Public Key: {public\_key}")

        print(f"Private Key: {private\_key}")

        msg = input("Enter the message: ")

        encrypted\_message = encrypt(public\_key, msg)

        print("\nEncrypted message:", encrypted\_message)

        decrypted\_message = decrypt(private\_key, encrypted\_message)

        print("Decrypted message:", decrypted\_message)

    except ValueError as e:

        print(e)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Output:

