Ex.No: 2 **Implementation of Hill and RSA Cipher.** Name : Shaurya

Date : 02.01.25 Reg.No : 22BAI1173

Aim

To implement the Hill and RSA Cipher.

Source Code

Hill Cipher

Python

```python
def generateKeyMatrix(n, key):
    k = 0
    keyMatrix = [[ ord(key[j*n+i]) % 65 for i in range(n) ] for j in range(n)]
    return keyMatrix

def encrypt(cipherMatrix, keyMatrix, msgVctr, n):
    for i in range(n):
        for j in range(1):
            cipherMatrix[i][j] = 0
            for x in range(n):
                cipherMatrix[i][j] += (keyMatrix[i][x] * msgVctr[x][j])

            cipherMatrix[i][j] = cipherMatrix[i][j] % 26

def HillCipher(msg, key):
    n = len(msg)
    keyMatrix = generateKeyMatrix(n, key)
    msgVctr = [[ord(msg[i]) % 65] for i in range(n)]
    cipherMatrix = [[0] for _ in range(n)]

    encrypt(cipherMatrix, keyMatrix, msgVctr, n)

    CipherText = []
    for i in range(n):
        CipherText.append(chr(cipherMatrix[i][0] + 65))

    print("Ciphertext:", "".join(CipherText))

n = int(input("Length of message: "))
msg = input("Message: ")
key = input("Key of length square of message: ")

HillCipher(msg, key)
```

RSA Cipher

Python

```
def gcd(a, b):
    while b:
        a, b = b, a % b
    return a

def findE(phi):
    e = 2
    while e < phi:
        if gcd(e, phi) == 1:
            return e
        e += 1
    return -1

def modInverse(e, phi):
    for d in range(1, phi):
        if (e * d) % phi == 1:
            return d
    return -1

def encryptRSA(message, e, n):
    messageInt = 0
    for char in message:
        messageInt = messageInt*10 + ord(char)%65

    cipher = (messageInt ** e) % n
    return cipher

p = int(input("Enter a prime number p: "))
q = int(input("Enter another prime number q: "))
n = p * q
phi = (p - 1) * (q - 1)

e = findE(phi)
d = modInverse(e, phi)

print(f"Public key (e, n): ({e}, {n})")
print(f"Private key (d, n): ({d}, {n})")

message = input("Enter the message to encrypt: ")
cipher = encryptRSA(message, e, n)
print("Encrypted message:", cipher)
```

Sample Input and Output

Input - Hill Cipher

Length of message: 3
Message: SKV
Key of length square of message: ACBDEGFHI

Output

Ciphertext: PMQ

```
Length of message: 3
Message: SKV
Key of length square of message: ACBDEGFHI
Ciphertext: PMQ
```

Input - RSA Cipher

Enter a prime number p: 23
Enter another prime number q: 11
Enter the message to encrypt: SKV

Output

Public key (e, n): (3, 253)
Private key (d, n): (147, 253)
Encrypted message: 233

```
Enter a prime number p: 23
Enter another prime number q: 11
Public key (e, n): (3, 253)
Private key (d, n): (147, 253)
Enter the message to encrypt: SKV
Encrypted message: 233
```

Solved Numericals

Hill Cipher

MSG: 'SKV' KEY= 'ACBDEGFHI'

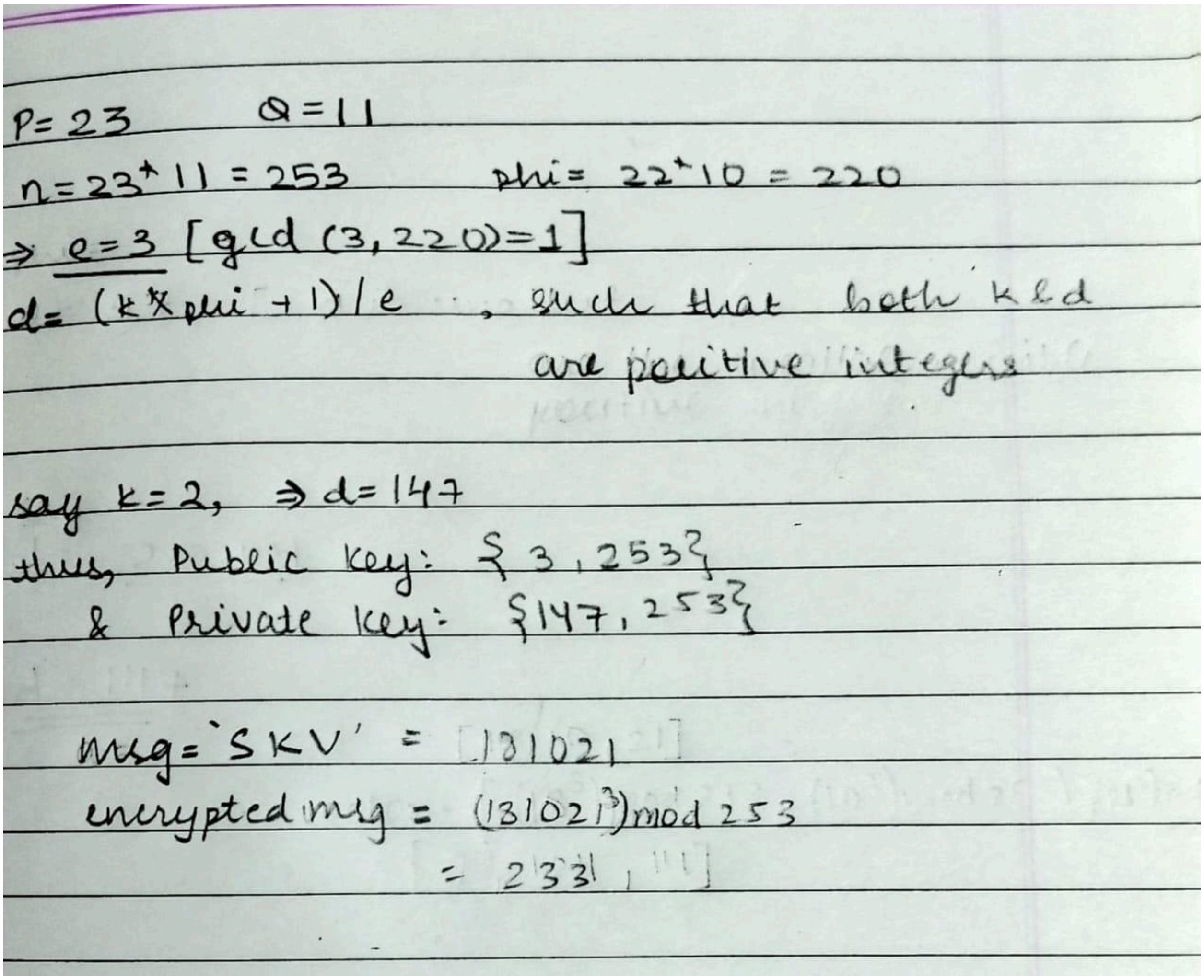
$$\text{KEY MATRIX} = \begin{bmatrix} 0 & 2 & 1 \\ 3 & 4 & 6 \\ 5 & 7 & 8 \end{bmatrix} \qquad \text{MSG VECTOR} = \begin{bmatrix} 18 \\ 10 \\ 21 \end{bmatrix}$$

$$\text{enciphered vector} = \begin{bmatrix} 0 & 2 & 1 \\ 3 & 4 & 6 \\ 5 & 7 & 8 \end{bmatrix} \times \begin{bmatrix} 18 \\ 10 \\ 21 \end{bmatrix} \pmod{26}$$

$$= \begin{bmatrix} 41 \\ 220 \\ 328 \end{bmatrix} \pmod{26} = \begin{bmatrix} 15 \\ 12 \\ 16 \end{bmatrix}$$

which corresponds to PMQ.

RSA Cipher



P = 23   Q = 11

n = 23 * 11 = 253   phi = 22 * 10 = 220

⇒ e = 3 [gcd (3, 220) = 1]

d = (k * phi + 1) / e , such that both k & d are positive integers

say k = 2, ⇒ d = 147

thus, Public key: {3, 253}

& Private key: {147, 253}

msg = 'SKV' = [18 10 21]

encrypted msg = (18 10 21^3) mod 253

= 2 3 3 [illegible]

Result

Thus we have implemented the Hill and RSA Ciphers.