**Practical-4**

**Aim : Alice wants to send some confidential information to Bob over a secure network. Prepare a key matrix for the given key and apply encryption on the plain text (key is your surname & plain text is your name).**

import string

def prepare\_key(key):

    key = key.upper().replace('J', 'I')

    unique\_chars = sorted(set(key), key=key.find)

    alphabet = list(string.ascii\_uppercase.replace('J', ''))

    matrix = unique\_chars + [letter for letter in alphabet if letter not in unique\_chars]

    playfair\_matrix = [matrix[i:i+5] for i in range(0, 25, 5)]

    return playfair\_matrix

def find\_position(char, matrix):

    for i, row in enumerate(matrix):

        if char in row:

            return i, row.index(char)

    return -1, -1

def process\_pair(char1, char2, matrix, operation):

    row1, col1 = find\_position(char1, matrix)

    if row1 == -1:

        row1, col1 = find\_position('X', matrix)

    row2, col2 = find\_position(char2, matrix)

    if row2 == -1:

        row2, col2 = find\_position('X', matrix)

    if row1 == row2:

        return matrix[row1][(col1 + operation) % 5] + matrix[row2][(col2 + operation) % 5]

    elif col1 == col2:

        return matrix[(row1 + operation) % 5][col1] + matrix[(row2 + operation) % 5][col2]

    else:

        return matrix[row1][col2] + matrix[row2][col1]

def encrypt(plaintext, matrix):

    ciphertext = ""

    plaintext = plaintext.upper().replace('J', 'I').replace(' ', '')

    pairs = [(plaintext[i], plaintext[i + 1] if i + 1 < len(plaintext) else 'X') for i in range(0, len(plaintext), 2)]

    for pair in pairs:

        ciphertext += process\_pair(pair[0], pair[1], matrix, 1)

    return ciphertext, pairs

def decrypt(ciphertext, matrix):

    plaintext = ""

    pairs = [(ciphertext[i], ciphertext[i + 1]) for i in range(0, len(ciphertext), 2)]

    for pair in pairs:

        plaintext += process\_pair(pair[0], pair[1], matrix, -1)

    return plaintext, pairs

key\_input = input("Enter the key: ")

plaintext\_input = input("Enter the plaintext: ")

playfair\_key\_matrix = prepare\_key(key\_input)

encrypted\_text\_output, encryption\_pairs\_output = encrypt(plaintext\_input, playfair\_key\_matrix)

print("\nEncryption Output:")

print(f"Plaintext Pairs: {encryption\_pairs\_output}")

print(f"Encrypted Text: {encrypted\_text\_output}")

print("\nKey Matrix:")

for row in playfair\_key\_matrix:

    print(row)

encrypted\_text\_input = input("\nEnter the encrypted text for decryption: ")

decrypted\_text\_output, decryption\_pairs\_output = decrypt(encrypted\_text\_input, playfair\_key\_matrix)

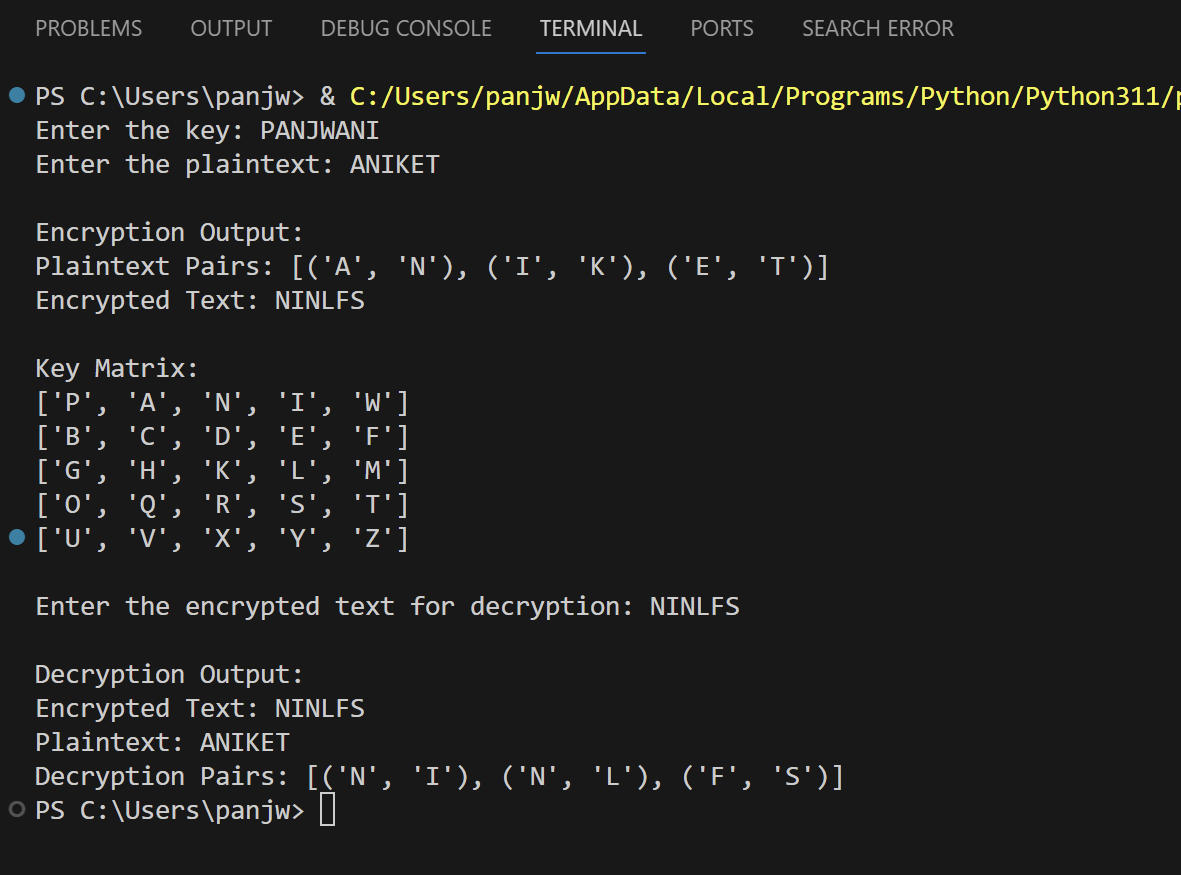
print("\nDecryption Output:")

print(f"Encrypted Text: {encrypted\_text\_input}")

print(f"Plaintext: {decrypted\_text\_output}")

print(f"Decryption Pairs: {decryption\_pairs\_output}")

**Output: -**

****