1. **Assign2.py**

import math

def encryptMessage(msg):

    cipher = ""

    k\_indx = 0

    msg = msg.replace(" ", "#")

    msg\_len = len(msg)

    msg\_lst = list(msg)

    key\_lst = sorted(list(key))

    col = len(key)

    row = int(math.ceil(msg\_len / col))

    fill\_null = int((row \* col) - msg\_len)

    msg\_lst.extend('\_' \* fill\_null)

    matrix = [msg\_lst[i: i + col] for i in range(0, len(msg\_lst), col)]

    key\_order = sorted(range(len(key)), key=lambda x: key[x])

    for i in range(col):

        curr\_idx = key\_order[i]

        cipher += ''.join([row[curr\_idx] for row in matrix])

    return cipher

def decryptMessage(cipher):

    msg = ""

    k\_indx = 0

    msg\_indx = 0

    msg\_len = len(cipher)

    msg\_lst = list(cipher)

    col = len(key)

    row = int(math.ceil(msg\_len / col))

    key\_lst = sorted(list(key))

    dec\_cipher = []

    for \_ in range(row):

        dec\_cipher += [[None] \* col]

    key\_order = sorted(range(len(key)), key=lambda x: key[x])

    for k in range(col):

        curr\_idx = key\_order[k]

        for j in range(row):

            dec\_cipher[j][curr\_idx] = msg\_lst[msg\_indx]

            msg\_indx += 1

    msg = ''.join(sum(dec\_cipher, []))

    msg = msg.replace("\_", "")

    msg = msg.replace("#", " ")

    return msg

msg = input("Enter the Plain Text: ")

key = input("Enter the Key: ")

cipher = encryptMessage(msg)

print("Encrypted Message:", cipher)

print("Decrypted Message:", decryptMessage(cipher))

1. **Output**

