**PRACTICAL NO. 1**

**Q1)** Write program to implement the following Substitution cipher techniques:

1. **Caeser Cipher**

**SOURCE CODE :**

def encrypt(string, shift):

    cipher = ''

    for char in string:

        if char == ' ':

            cipher = cipher + char

        elif char.isupper():

            cipher = cipher + chr((ord(char) + shift - 65) % 26 + 65)

        else:

            cipher = cipher + chr((ord(char) + shift - 97) % 26 + 97)

    return cipher

def decrypt(string, shift):

    cipher = ''

    for char in string:

        if char == ' ':

            cipher = cipher + char

        elif char.isupper():

            cipher = cipher + chr((ord(char) + (26-shift) - 65) % 26 + 65)

        else:

            cipher = cipher + chr((ord(char) + (26-shift) - 97) % 26 + 97)

    return cipher

text = input("Enter String : ")

s = int(input("enter Shift Number : "))

option = int(input("1. For Encrypt \n2. For Decrypt\n Enter Your choice : "))

print("Original String : ", text)

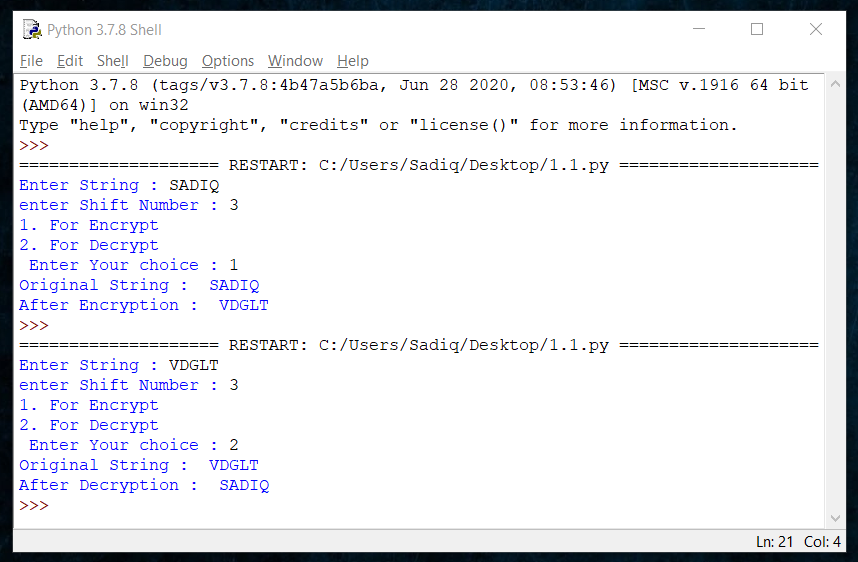
if( option == 1):

    print("After Encryption : ", encrypt(text, s))

else:

    print("After Decryption : ", decrypt(text, s))

**OUTPUT :**

****

1. **Monoalphabetic Cipher**

**SOURCE CODE :**

import random

alpha = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

key = ''.join(random.sample(alpha,len(alpha)))#generates a random shuffle of alphabets

def encrypt(encrypt\_msg):

final\_ans = ""

for char in encrypt\_msg:#iterates over every character in text to be encrypted

num = alpha.find(char)#num = index of character

final\_ans = final\_ans + key[num]#append new char from random shuffle of alphabets

print(final\_ans)

def decrypt(decrypt\_msg):

final\_ans = ""

for char in decrypt\_msg:#iterates over every character in text to be encrypted

num = key.find(char)#num = index of character

final\_ans = final\_ans + alpha[num]#append new char from alphabets

print(final\_ans)

while True:#Menu driven While loop; goes on untill EXIT option is not selected or program is not killed

n = int(input("Enter value:: \n1) To Encrypt Text:: \n2) To Decrypt Text:: \n3) See Key:: \n4) To Exit \n"))

if (n == 1):

encrypt\_msg = str(input("Enter Text:: "))

encrypt(encrypt\_msg.upper())

elif (n == 2):

decrypt\_msg = str(input("Enter Crypt Text:: "))

decrypt(decrypt\_msg.upper())

elif (n==3):

print(key)

elif (n==4):

break

else:

print("Invalid Input; Enter Again!!")

**OUTPUT :**

