**CNS LAB**

**Name: Samrat Vishwas Jadhav PRN: 2020BTECS00006**

**Assignment 1**

**Aim - To encrypt the given plain text using Caesar Cipher and then decrypt it to get plain text again.**

It is substitution cipher, i.e., each letter of a given text is replaced by a letter with a fixed number of positions down the alphabet

Code:

import enchant

d = enchant.Dict("en\_US")

def encrypt(text, shift):

    encrypted\_txt = ""

    for i in range(len(text)):

        char = text[i]

*# Checking for valid input text*

        if ord(char) in range(65,91) or ord(char) in range(97,123):

*# Encrypting Uppercase characters (A=65, B=66, ...)*

            if (char.isupper()):

                encrypted\_txt += chr((ord(char) + shift-65) % 26 + 65)

*# Encrypting Lowercase characters (a=97, b=98, ...)*

            else:

                encrypted\_txt += chr((ord(char) + shift-97) % 26 + 97)

        elif char == " ":

            encrypted\_txt += " "

        else:

            encrypted\_txt = f"Invalid character {char} in input string!\nOnly Alphabets [a-z][A-Z] are allowed"

            break

    return encrypted\_txt

def decrypt(text, shift):

    decrypted\_txt = ""

    shift = 26-shift

    for i in range(len(text)):

        char = text[i]

*# Checking for valid input text*

        if ord(char) in range(65,91) or ord(char) in range(97,123):

*# Decrypting Uppercase characters*

            if (char.isupper()):

                decrypted\_txt += chr((ord(char) + shift-65) % 26 + 65)

*# Decrypting Lowercase characters*

            else:

                decrypted\_txt += chr((ord(char) + shift-97) % 26 + 97)

        elif char == " ":

                decrypted\_txt += " "

        else:

            decrypted\_txt = f"Invalid character {char} in input string!\nOnly Alphabets [a-z][A-Z] are allowed"

            break

    return decrypted\_txt

def is\_english\_word(word):

    return d.check(word)

*# Main Program*

c = int(input("What do you want to perform?\n1. Encryption\n2. Decryption\n"))

if c==1:

*# Encryption*

    txt = str(input("Enter the text to be encrypted: "))

    shift = int(input("Enter the shift: "))

*# Checking for valid shift value*

    if shift in range(1, 26):

        print(f"Ciphertext is:\n{encrypt(txt, shift)}")

    else:

        print("Invalid Shift value !!\nValue must be in between 1-25")

elif c==2:

*# Decryption*

    txt = str(input("Enter the text to be decrypted: "))

    shift\_known = str(input("Do you know the shift value [y/n] ?  "))

    if shift\_known == 'y':

        shift = int(input("Enter the shift: "))

*# Checking for valid shift value*

        if shift in range(1, 26):

            print(f"Decrypted text is:\n{decrypt(txt, shift)}")

        else:

            print("Invalid Shift value !!\nValue must be in between 1-25")

    elif shift\_known == 'n':

        dec\_txt = []

        for shift in range(1, 26):

            dec\_txt.append(decrypt(txt, shift))

        is\_found = False

        for snt in dec\_txt:

            correct\_words = 0

            words = snt.split()

            for word in words:

                if is\_english\_word(word):

                    correct\_words += 1

            if correct\_words == len(words):

                is\_found = True

                print(f"Most appropriate decrypted text is:\n{snt}\nShift used = {dec\_txt.index(snt)+1}")

        if not is\_found:

            print("Unable to find the appropriate match!\nList of all possible decrypted texts is:")

            for i, snt in enumerate(dec\_txt):

                print(f"{i+1}. {snt}")

    else:

        print("Invalid input!")

else:

    print("Invalid input!")

**Output:**

