**Cryptography and Network Security Lab**

**Name: Rushikesh R. Ware**

**PRN: 2020BTECS00012**

**Batch: B2**

**CAESAR CIPHER ALGORITHM**

**Aim:**

To encrypt the plain text using Caesar cipher and decrypt to get plain text from encrypted text.

**Theory:**

The Caesar Cipher technique is one of the earliest and simplest encryption technique. It’s a substitution cipher. Substitution cipher is nothing but a technique in which a letter is replaced by any other letter. The method is named after Julius Caesar, who used it in his private correspondence.

**Code:**

**Encryption:**

def encrypt(plain\_text, shift):

    caesar\_cipher = ""

    for char in plain\_text:

        if char == ' ':

            caesar\_cipher += ' '

        elif char.isupper():

            caesar\_cipher += chr((ord(char) + shift - 65) % 26 + 65)

        else:

            caesar\_cipher += chr((ord(char) + shift - 97) % 26 + 97)

    return caesar\_cipher

def main():

    plain\_text = input("enter plain text: ")

    print()

    shift = int(input("enter value of shift: "))

    print()

    caesar\_cipher = encrypt(plain\_text, shift)

    print("The Caesar cipher encrypted text is:")

    print(caesar\_cipher)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Decryption:**

import nltk

from nltk.corpus import words

nltk.download("words")

english\_words = set(words.words())

def decrypt(ciphertext, s):

    result = ""

    for char in ciphertext:

        if char == " ":

            result += " "

        elif char.isupper():

            result += chr((ord(char) - s - 65) % 26 + 65)

        else:

            result += chr((ord(char) - s - 97) % 26 + 97)

    return result

def main():

    ciphertext = input("Enter encrypted text: ")

    print()

    shift\_key=0

    decrypted\_text\_store=""

    for s in range(1, 27):

        decrypted\_text = decrypt(ciphertext, s)

        decrypted\_words = decrypted\_text.split()

        # print(decrypted\_words)

        is\_valid = all(word.lower() in english\_words for word in decrypted\_words)

        if is\_valid:

            shift\_key=s

            decrypted\_text\_store=decrypted\_text

            break

    if s and decrypted\_text\_store:

        print("Valid decryption with shift key", shift\_key)

        print("Decrypted Text:", decrypted\_text)

    else:

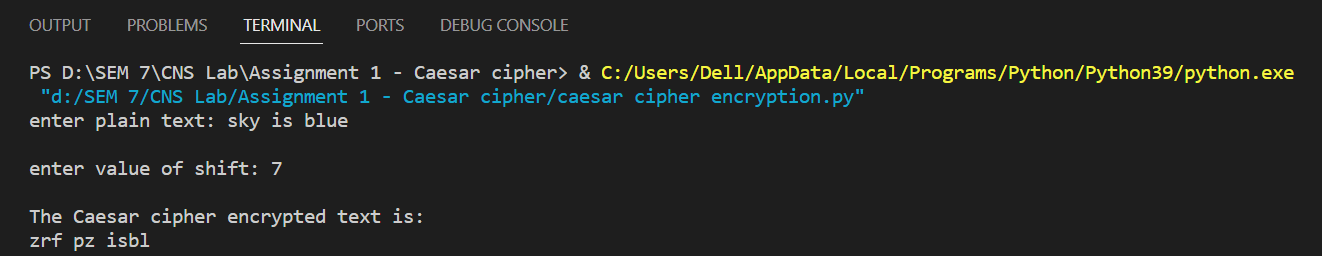
            print("Decryption Unsuccessful")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Output:**

**Encryption:**



**Decryption:**

