**NIS\_LAB\_1\_Assignment**

**CE\_055**

**Aim :-** Aim is to develop the simple Caesar Cipher and Substitution Cipher.

1. Additive Cipher or Caesar Cipher.

Code:-

'''

Author : Dhruv B Kakadiya

'''

# Additive Cipher Algorithm or Ceasor Cipher Algorithm

mod = 26

# encryption function for encrypt the plain text into the cipher text

def encryption (plain\_text, key):

    encrypted\_text = ""

    for letter in plain\_text:

        if (letter.isupper()):

            encrypted\_text += chr((ord(letter) - 65 + key) % mod + 65)

        elif (letter.islower()):

            encrypted\_text += chr((ord(letter) - 97 + key) % mod + 97)

        else:

            encrypted\_text += letter

    return encrypted\_text

# depcryption function for decode the encrypted text

def decryption (encrypted\_text, key):

    decrypted\_text = ""

    for letter in encrypted\_text:

        if (letter.isupper()):

            decrypted\_text += chr((ord(letter) - 65 - key) % mod + 65)

        elif (letter.islower()):

            decrypted\_text += chr((ord(letter) - 97 - key) % mod + 97)

        else:

            decrypted\_text += letter

    return decrypted\_text

# crypt analysis function for attackers to find the appropriate text matching

def crypt\_analysis (encrypted\_text):

    try\_match\_text\_list = []

    for key in range(1, 26):

        try\_match\_text = ""

        for letter in encrypted\_text:

            if (letter.isupper()):

                try\_match\_text += chr((ord(letter) - 65 - key) % mod + 65)

            else:

                try\_match\_text += chr((ord(letter) - 97 - key) % mod + 97)

        try\_match\_text\_list.append(try\_match\_text)

    return try\_match\_text\_list

if \_\_name\_\_ == "\_\_main\_\_":

    plain\_text = input("\nEnter the plain text : ")

    key = int(input("\nEnter the encryption key : "))

    cipher\_text = encryption(plain\_text, key)

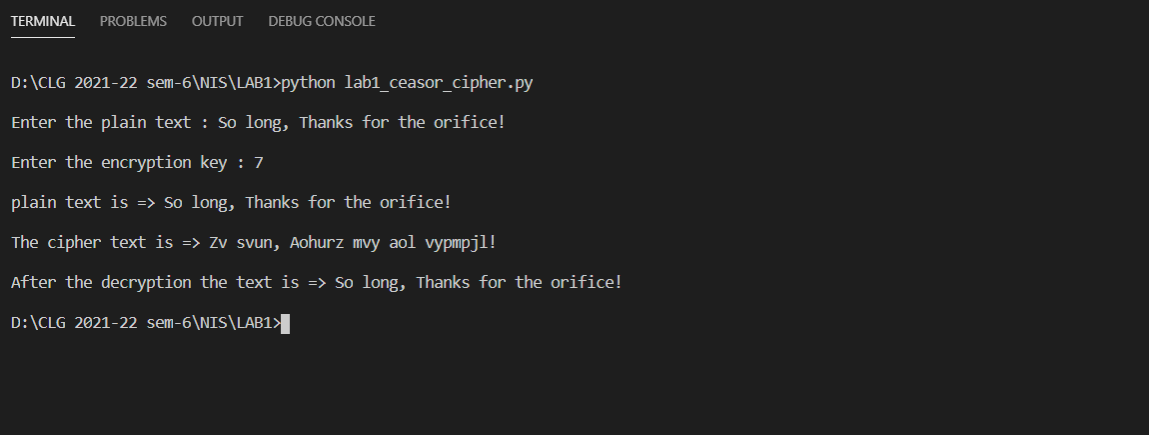
    print(f"\nplain text is => {plain\_text}")

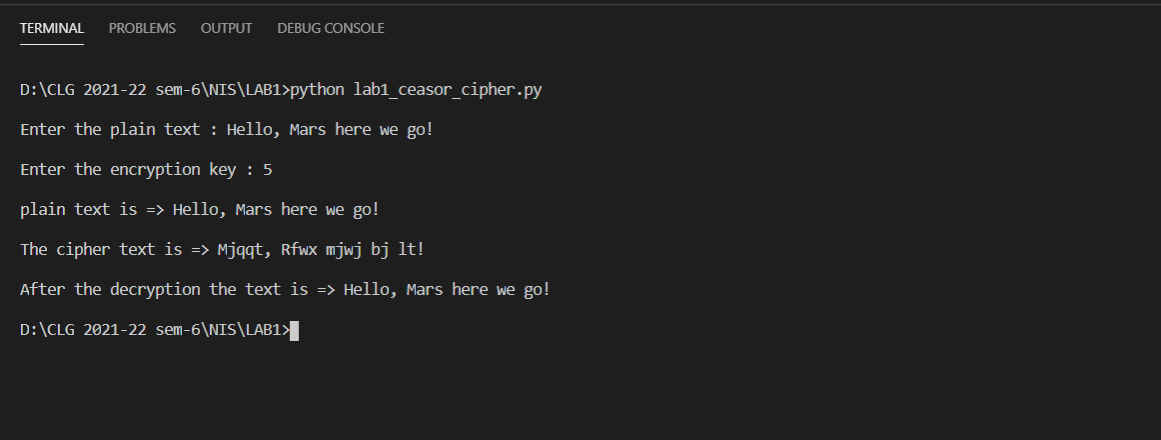
    print(f"\nThe cipher text is => {cipher\_text}")

    decrypted\_text = decryption(cipher\_text, key)

    print(f"\nAfter the decryption the text is => {decrypted\_text}")

Output:





1. Monoalphabetic Cipher or Substitution Cipher.

Code :

'''

Author : Dhruv B Kakadiya

'''

# Substitution Cipher Algorithm or Monoalphabetic Cipher Algorithm

Organized\_key = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

shuffled\_key = "QWERTYUIOPASDFGHJKLZXCVBNM"

def encrypt\_decrypt\_text(input\_text, key, mode = None):

    converted\_text = ""

    org = Organized\_key

    shuff = key

    if (mode == "decryption"):

        org, shuff = shuff, org

    for letter in input\_text:

        if (letter.upper() in org):

            if (letter.isupper()):

                converted\_text += shuff[org.find(letter.upper())].upper()

            else:

                converted\_text += shuff[org.find(letter.upper())].lower()

        else:

            converted\_text += letter

    return converted\_text

if \_\_name\_\_ == "\_\_main\_\_":

    input\_text = input("\nEnter the text : ")

    print("\n1. encryption\n2. decryption")

    mode = input("\nchoose mode : ")

    if (mode == "encryption"):

        result = encrypt\_decrypt\_text(input\_text, shuffled\_key, mode)

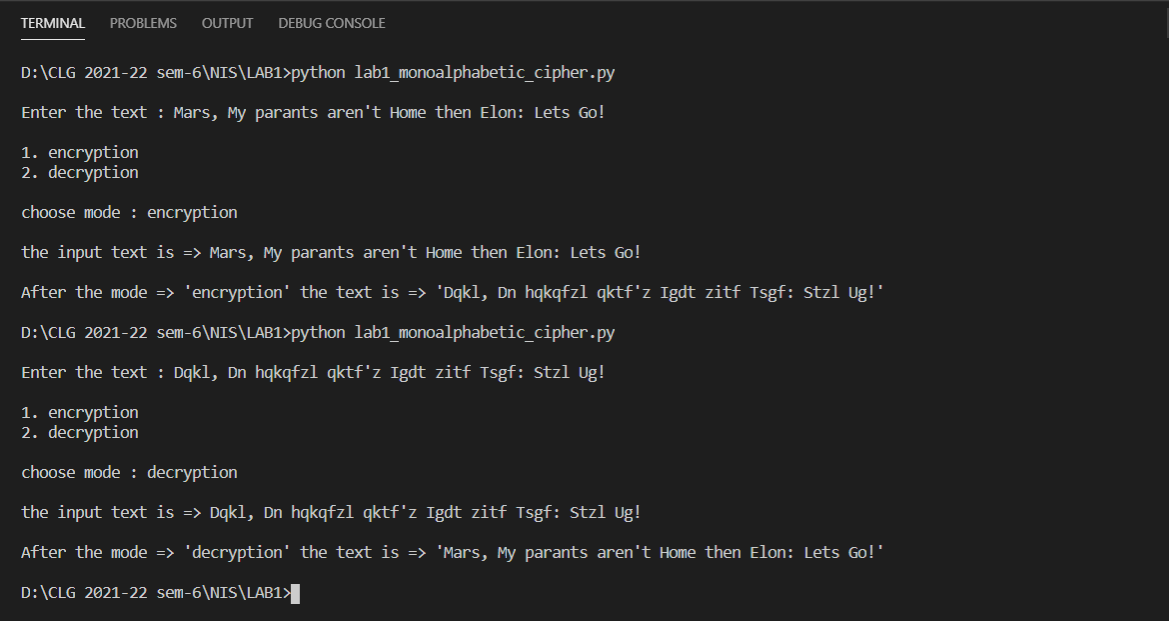
    else:

        result = encrypt\_decrypt\_text(input\_text, shuffled\_key, mode)

    print(f"\nthe input text is => {input\_text}")

    print(f"\nAfter the mode => '{mode}' the text is => '{result}'")

Output:



Description:

1. Additive Cipher / Caesar Cipher :

* This is a most common algorithm for encryption purpose and it is very easily to decrypt it using brute force method.
* In this algorithm we have chosen key as a number and define every alphabet with respect to number like a-1, b-2, c-3…, and put the encrypted value of each and every letter in some string according to case sensitively.
* The formula for this algorithm is

E(x) = (x + n) mod 26

D(x) = (x - n) mod 26

* That is why this algorithm is not strong. because there are only 26 possibilities for key.

1. Monoalphabetic Cipher / Substitution Cipher:

* In this algorithm the key is randomly shuffle alphabets that’s why we assign each and every character of plain text with the unique alphabet using index inside the key.
* So, the first letter of plain text has 26 possibilities next has 25 and so on, that’s why there is 26! Possibilities to decrypt the encrypted text.
* This Algorithm is much better than Caesar cipher but this algo is not too Strong.