NIS_LAB_1_Assignment

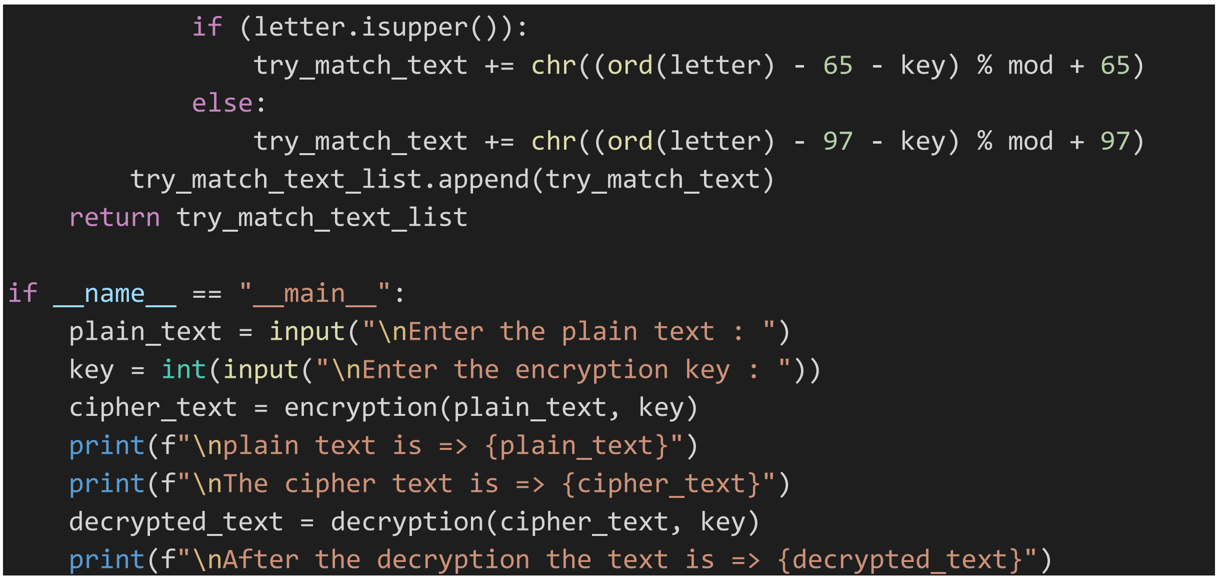
CE_055

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

1. Additive Cipher or Caesar Cipher.

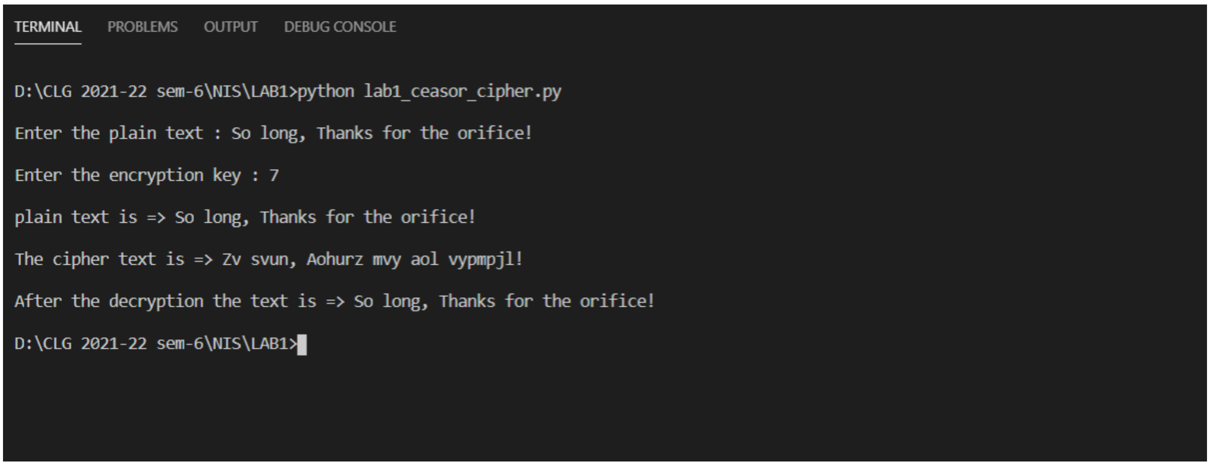
Code:-

```
'''
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:
```

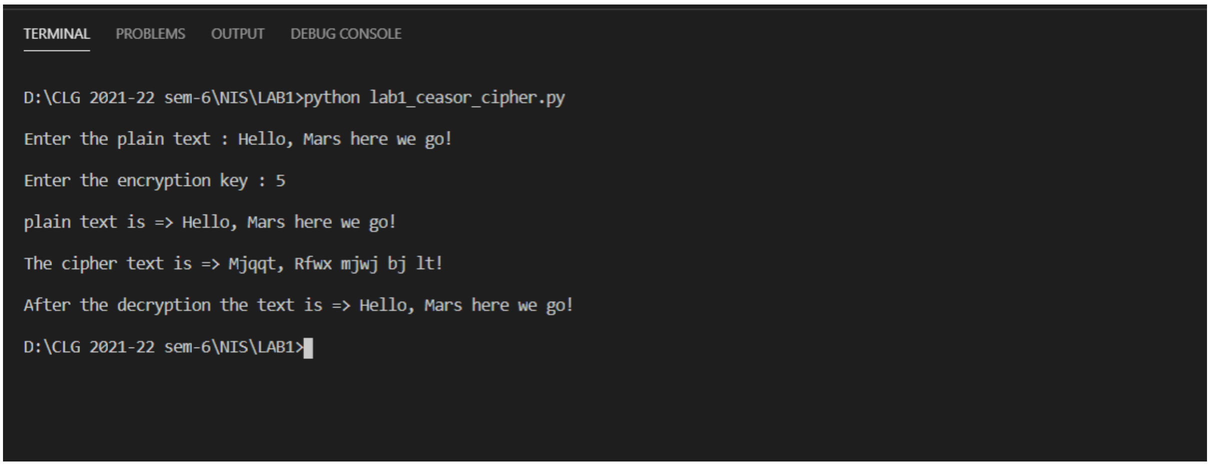


```
            if (letter.isupper()):
                try_match_text += chr((ord(letter) - 65 - key) % mod + 65)
            else:
                try_match_text += chr((ord(letter) - 97 - key) % mod + 97)
        try_match_text_list.append(try_match_text)
    return try_match_text_list

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:



```
D:\CLG 2021-22 sem-6\NIS\LAB1>python lab1_ceasor_cipher.py

Enter the plain text : So long, Thanks for the orifice!

Enter the encryption key : 7

plain text is => So long, Thanks for the orifice!

The cipher text is => Zv svun, Aohurz mvy aol vypmpjl!

After the decryption the text is => So long, Thanks for the orifice!

D:\CLG 2021-22 sem-6\NIS\LAB1>
```



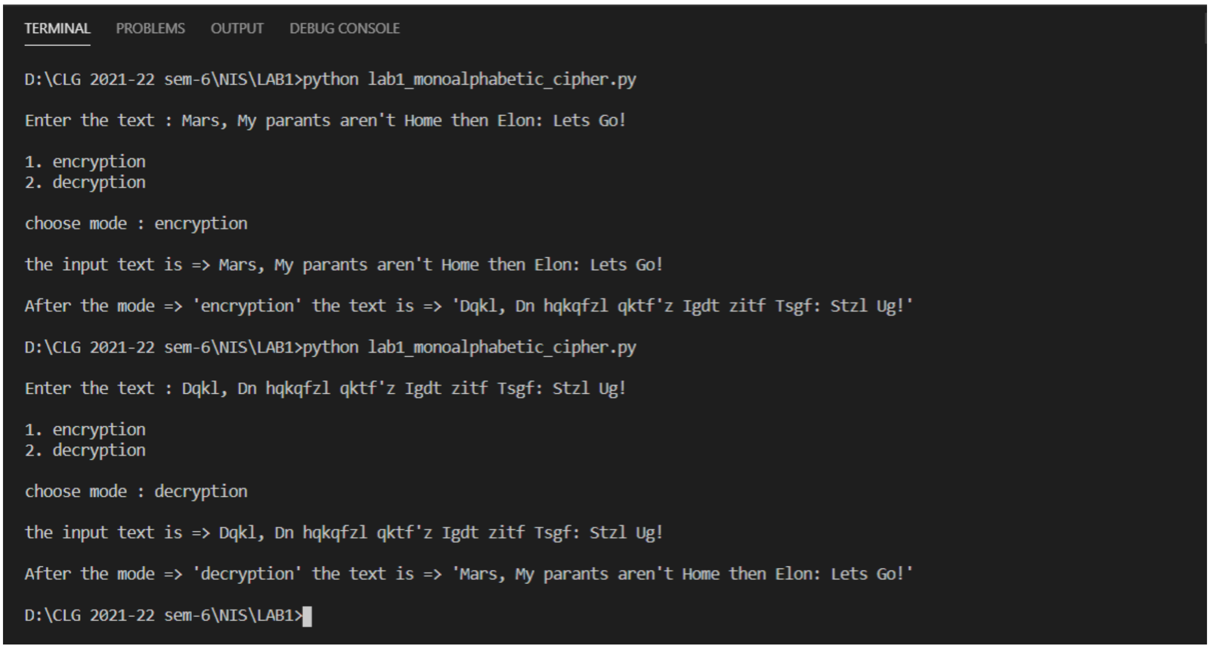
```
D:\CLG 2021-22 sem-6\NIS\LAB1>python lab1_ceasor_cipher.py

Enter the plain text : Hello, Mars here we go!

Enter the encryption key : 5

plain text is => Hello, Mars here we go!

The cipher text is => Mjqqt, Rfwx mjwj bj lt!

After the decryption the text is => Hello, Mars here we go!

D:\CLG 2021-22 sem-6\NIS\LAB1>
```

2. Monoalphabetic Cipher or Substitution Cipher.

Code :

```python
'''
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:



```
D:\CLG 2021-22 sem-6\NIS\LAB1>python lab1_monoalphabetic_cipher.py

Enter the text : Mars, My parants aren't Home then Elon: Lets Go!

1. encryption
2. decryption

choose mode : encryption

the input text is => Mars, My parants aren't Home then Elon: Lets Go!

After the mode => 'encryption' the text is => 'Dqkl, Dn hqkqfzl qktf'z Igdt zitf Tsgf: Stzl Ug!'

D:\CLG 2021-22 sem-6\NIS\LAB1>python lab1_monoalphabetic_cipher.py

Enter the text : Dqkl, Dn hqkqfzl qktf'z Igdt zitf Tsgf: Stzl Ug!

1. encryption
2. decryption

choose mode : decryption

the input text is => Dqkl, Dn hqkqfzl qktf'z Igdt zitf Tsgf: Stzl Ug!

After the mode => 'decryption' the text is => 'Mars, My parants aren't Home then Elon: Lets Go!'

D:\CLG 2021-22 sem-6\NIS\LAB1>
```

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.

2. 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.