Perform encryption, decryption using the following substitution techniques [Ceaser cipher]

Ex. No : 1a

Date :

Aim:

To perform encryption and decryption using Ceaser Cipher.

Algorithm:

Step 1: Obtain the text for encryption /decryption

Step 2: Get input from the user to Encrypt/Decrypt

Step 3: Get the key from the user.

Step 4: Perform an encryption/decryption using key.

Step 5: Output the corresponding Plaintext/Cipher Text.

Source code:

```
def encrypt(text,key):
   result=""
   for i in range(len(text)):
      result += chr(((ord(text[i])-97+key)%26)+97)
   print("Encrypted text :",result)

def decrypt(text,key):
   result=""
   for i in range(len(text)):
      result += chr(((ord(text[i])-97-key)%26)+97)
   print("Decrypted text :",result)

def main():
   text = input("Enter the text for encrypt/decrypt : ")
```

```
    key = int(input("Enter the key : "))
    choice = int(input("1.Encrypt 2.Decrypt : "))
    if(choice==1):
        encrypt(text,key)
    else:
        decrypt(text,key)

main()
```

Output:

```
Enter the text for encrypt/decrypt : security
Enter the key : 12
1.Encrypt 2.Decrypt : 1
Encrypted text : eqogdufk
```

```
Enter the text for encrypt/decrypt : eqogdufk
Enter the key : 12
1.Encrypt 2.Decrypt : 2
Decrypted text : security
```

Result:

The Ceaser cipher encryption and decryption technique was executed successfully and output was verified.

Perform encryption, decryption using the following substitution techniques [Playfair cipher]

Ex. No : 1b

Date :

Aim:

To perform encryption and decryption using playfair Cipher.

Algorithm:

Step 1: Obtain the text for encryption /decryption

Step 2: Get input from the user to Encrypt/Decrypt

Step 3: Get the key from the user.

Step 4: Perform an encryption/decryption using key.

Step 5: Output the corresponding Plaintext/Cipher Text.

Source code:

```
class TwoWayDict(dict):
    def __len__(self):
       return dict.__len__(self) / 2
    def __setitem__(self, key, value):
        dict.__setitem__(self, key, value)
        dict.__setitem__(self, value, key)
d=TwoWayDict()
def fill_key_matrix(key):
            count=0
            for i in range(len(key)):
                        if(key[i] not in d.keys()):
                                    d[key[i]]=count
                                    count+=1
            for i in range(97,123,1):
```

```
        if(chr(i) not in d.keys() and chr(i)!='j'):
            d[chr(i)]=count
            count+=1
def normalize(text):
    result=text[0]
    for i in range(1,len(text)):
        if(text[i]==text[i-1]):
            result+="x"+text[i]
        else:
            result+=text[i]
    if(len(result)%2!=0):
        result+="x"
    return result
def encrypt(text,key):
    text=normalize(text)
    result=""
    for i in range(0,len(text),2):
        row1 = d[text[i]]//5 #integer division
        col1=d[text[i]]%5
        row2=d[text[i+1]]//5
        col2=d[text[i+1]]%5
        if(row1==row2):
            col1=(col1+1)%5
            col2=(col2+1)%5
            result+=d[row1*5+col1]
            result+=d[row2*5+col2]
        elif(col1==col2):
            row1=(row1+1)%5
            row2=(row2+1)%5
            result+=d[row1*5+col1]
            result+=d[row2*5+col2]
        else:
            result+=d[row1*5+col2]
            result+=d[row2*5+col1]
    print("Encrypted Text : ", result)

def decrypt(text,key):
    result=""
    for i in range(0,len(text),2):
        row1 = d[text[i]]//5 #integer division
        col1=d[text[i]]%5
        row2=d[text[i+1]]//5
        col2=d[text[i+1]]%5
        if(row1==row2):
            col1=(col1-1)%5
            col2=(col2-1)%5
            result+=d[row1*5+col1]
            result+=d[row2*5+col2]
        elif(col1==col2):
            row1=(row1-1)%5
            row2=(row2-1)%5
            result+=d[row1*5+col1]
            result+=d[row2*5+col2]
        else:
            result+=d[row1*5+col2]
            result+=d[row2*5+col1]

    print("Decrypted Text : ",result)
def main():
    text = input("Enter Text to encrypt/decrypt : ")
    text=text.replace("j","i")
    key= input("Enter key : ")
    key.replace("j","i")
    fill_key_matrix(key)
    choice = int(input("Enter 1.Encrypt 2.Decrypt :  "))
    if(1==choice):
        encrypt(text,key)
    else:
        decrypt(text,key)
main()
```

Output:

```
Enter Text to encrypt/decrypt : technology
Enter key : cat
Enter 1.Encrypt 2.Decrypt :  1
Encrypted Text :  cgbeokmkhx
```

```
Enter Text to encrypt/decrypt : cgbeokmkhx
Enter key : cat
Enter 1.Encrypt 2.Decrypt :  2
Decrypted Text :  technology
```

Result:

The playfair cipher encryption and decryption technique was executed successfully and output was verified.

Perform encryption, decryption using the following substitution techniques

[Hill Cipher]

Ex. No : 1c

Date :

Aim:

To perform encryption and decryption using Hill Cipher.

Algorithm:

Step 1: Obtain the text for encryption /decryption

Step 2: Get input from the user to Encrypt/Decrypt

Step 3: Get the key from the user.

Step 4: Perform an encryption/decryption using key.

Step 5: Output the corresponding Plaintext/Cipher Text.

Source code:

```
import math
import numpy as np
from sympy import Matrix
def normalize(text,bl):
        text+= "x"*(  (-(len(text)%bl))%bl)
        return text
def aton(n):
        return ord(n)-97
def encrypt(text,key):
        bl=int(math.sqrt(len(key)))
        text=normalize(text,bl)
        key = np.array(list(map(aton,list(key)))).reshape(bl,bl)
        ct=""
        for i in range(0,len(text)//bl,bl):
```

```
        l=list(map(aton,text[i:i+bl:1]))
        l=np.array(l).reshape(bl,1)
        l=np.dot(key,l)
        l=l%26
        for j in range(bl):
            ct+=chr(l[j][0]+97)
    print(ct)
def decrypt(text,key):
    bl=int(math.sqrt(len(key)))
    #text=normalize(text,bl)
    key = np.array(list(map(aton,list(key)))).reshape(bl,bl)
    key= np.array(Matrix(key).inv_mod(26))
    pt=""
    for i in range(0,len(text)//bl,bl):
        l=list(map(aton,text[i:i+bl:1]))
        l=np.array(l).reshape(bl,1)
        l=np.dot(key,l)
        l=l%26
        for j in range(bl):
            pt+=chr(l[j][0]+97)
    print(pt)
def main():
    text = input("Enter Text to encrypt/decrypt : ")
    key=input("Enter Key : ")
    choice = int(input("Enter 1.Encrypt 2.Decrypt :  "))
    if(1==choice):
        encrypt(text,key)
    else:
        decrypt(text,key)
main()
```

Output:

```
Enter Text to encrypt/decrypt : ACT

Enter key : GYBNQKURP

Enter 1.Encrypt 2.Decrypt :  1

Encrypted Text :  POH
```

```
Enter Text to encrypt/decrypt : POH

Enter key : GYBNQKURP

Enter 1.Encrypt 2.Decrypt :  2

Decrypted Text :  ACT
```

Result:

The Hill cipher encryption and decryption technique was executed successfully and output was verified.

Perform encryption, decryption using the following substitution techniques [Vignere Cipher]

Ex. No : 1d

Date :

Aim:

To perform encryption and decryption using VignereCipher.

Algorithm:

Step 1: Obtain the text for encryption /decryption

Step 2: Get input from the user to Encrypt/Decrypt

Step 3: Get the key from the user.

Step 4: Perform an encryption/decryption using key.

Step 5: Output the corresponding Plaintext/Cipher Text.

Source code:

```
def encrypt(text,key):
        result=""
        pos=0
        n=len(key)
        for i in range(len(text)):
                result+=  chr ( ( (ord(text[i])-97+ord(key[pos])- 97)%26 )+97)
                pos= (pos+1)%n
        print("Encrypted Text ",result)

def decrypt(text,key):
        result=""
        pos=0
        n=len(key)
        for i in range(len(text)):
```

```
        result+=  chr ( ( (ord(text[i])-ord(key[pos]) )%26 )+97)
        pos=(pos+1)%n
    print("Decrypted Text ",result)



def main():
    text = input("Enter Text to encrypt/decrypt : ")
    key=input("Enter key : ")
    choice = int(input("Enter 1.Encrypt 2.Decrypt :  "))
    if(1==choice):
        encrypt(text,key)
    else:
        decrypt(text,key)
main()
```

Output:

```
Enter Text to encrypt/decrypt : securitylab
Enter key : hat
Enter 1.Encrypt 2.Decrypt :  1
Encrypted Text  zevbrbayehb
```

```
Enter Text to encrypt/decrypt : zevbrbayehb
Enter key : hat
Enter 1.Encrypt 2.Decrypt :  2
Decrypted Text  securitylab
```

Result:

The Vignere cipher encryption and decryption technique was executed successfully and output was verified.