Experiment no. 2

Roll no = B-03

Aim: Cryptoanalysis or decoding playfair, vignere cipher using python.

Theory :-
Playfair cipher was first practical diagraph substitution cipher. Scheme was invented in 1854 by charles wheatstone but was named after lord Playfair who promoted use of cipher

In playfair cipher unlike traditional cipher we encrypt a pair of alphabets (diagraphs) instead of a single alphabet.

Encryption Algorithm.

step 1 – Generate key square (5x5)
Key square is a 5x5 grid of alphabets that acts as key for encrypting plaintext. Each of 25 alphabets must be unique & one letter of alphabet (usually J) is omited from table. If plaintext contains J then it is replaced by I.

Initial alphabets in key Square are unique alphabets of key in which they appear Folowed by remaining letters of alphabet in order.

step 2 - Algorithm to encrypt plain text is split into pairs of two letters. If there is odd number of lettes, a Z is added to last letter.

Decryption Algorithm

Step 1 - Generate key square at receiver's end
Key square is 5x5 grid of alphabets that acts as key for encrypting plain text.
Initial alphabets in key Square are unique alphabets of key in order in which they appear Followed by remaining letters of alphabet in order.

Step 2 - Algorithm to decrypt cipher text. Ciphertext is split into pairs of two letters.

Vigenere cipher

A method of encrypting alphabetic text.
Table consists of alphabets written out 26 times in different rows.
At different point in encryption process cipher uses different alphabet from one of rows.

Encryption

In Vignere table it is 26×26 matrix where rows is key & column is plain text. Each letter of plaintext & key are compared & there intersection results in ciphertext. Similar process in followed for other plaintext letters.

Decryption

Decryption is performed by giving column to key & finding corresponding position of ciphertext in this column using row's label as plaintext.

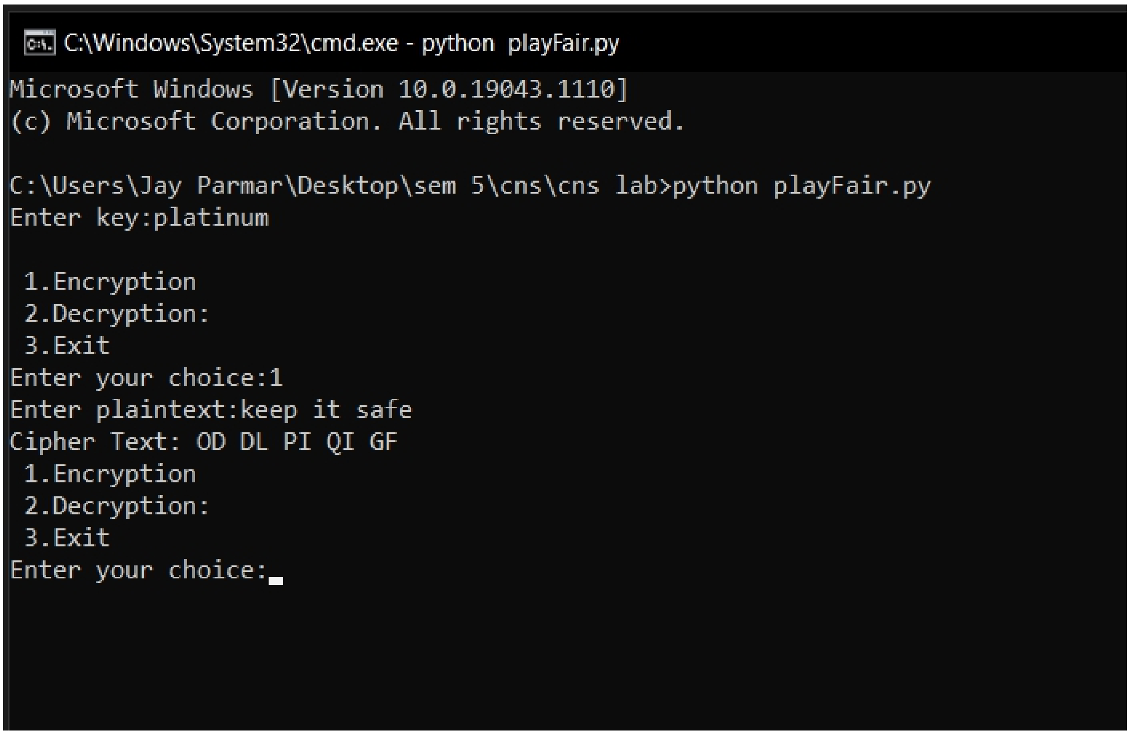
PLAYFAIR CIPHER

PROGRAM :

```
key=input("Enter key:")
key=key.replace(" ", "")
key=key.upper()
def matrix(x,y,initial):
    return [[initial for i in range(x)] for j in range(y)]


result=list()
for c in key: #storing key
    if c not in result:
        if c=='J':
            result.append('I')
        else:
            result.append(c)
flag=0
for i in range(65,91): #storing other character
    if chr(i) not in result:
        if i==73 and chr(74) not in result:
            result.append("I")
            flag=1
        elif flag==0 and i==73 or i==74:
            pass
        else:
            result.append(chr(i))
k=0
```

```
my_matrix=matrix(5,5,0) #initialize matrix
for i in range(0,5): #making matrix
    for j in range(0,5):
        my_matrix[i][j]=result[k]
        k+=1

def locindex(c): #get location of each character
    loc=list()
    if c=='J':
        c='I'
    for i ,j in enumerate(my_matrix):
        for k,l in enumerate(j):
            if c==l:
                loc.append(i)
                loc.append(k)
                return loc

def encrypt():  #Encryption
    msg=str(input("Enter plaintext:"))
    msg=msg.upper()
    msg=msg.replace(" ", "")
    i=0
    for s in range(0,len(msg)+1,2):
        if s<len(msg)-1:
            if msg[s]==msg[s+1]:
                msg=msg[:s+1]+'X'+msg[s+1:]
    if len(msg)%2!=0:
        msg=msg[:]+'X'
    print("Cipher Text:",end=' ')
    while i<len(msg):
        loc=list()
        loc=locindex(msg[i])
        loc1=list()
        loc1=locindex(msg[i+1])
        if loc[1]==loc1[1]:

print("{}{}".format(my_matrix[(loc[0]+1)%5][loc[1]],my_matrix[(loc1[0]+1)%5][l
oc1[1]]),end=' ')
        elif loc[0]==loc1[0]:

print("{}{}".format(my_matrix[loc[0]][(loc[1]+1)%5],my_matrix[loc1[0]][(loc1[1]
+1)%5]),end=' ')
        else:

print("{}{}".format(my_matrix[loc[0]][loc1[1]],my_matrix[loc1[0]][loc[1]]),end='
')
        i=i+2


def decrypt():  #decryption
    msg=str(input("ENTER CIPHER TEXT:"))
    msg=msg.upper()
    msg=msg.replace(" ", "")
    print("PLAIN TEXT:",end=' ')
    i=0
    while i<len(msg):
        loc=list()
        loc=locindex(msg[i])
        loc1=list()
        loc1=locindex(msg[i+1])
        if loc[1]==loc1[1]:
            print("{}{}".format(my_matrix[(loc[0]-1)%5][loc[1]],my_matrix[(loc1[0]-
1)%5][loc1[1]]),end=' ')
        elif loc[0]==loc1[0]:
            print("{}{}".format(my_matrix[loc[0]][(loc[1]-
1)%5],my_matrix[loc1[0]][(loc1[1]-1)%5]),end=' ')
        else:

print("{}{}".format(my_matrix[loc[0]][loc1[1]],my_matrix[loc1[0]][loc[1]]),end='
')
        i=i+2

while(1):
    print("\n 1.Encryption \n 2.Decryption: \n 3.Exit")
    choice=int(input("Enter your choice:"))
    if choice==1:
        encrypt()
    elif choice==2:
        decrypt()
    elif choice==3:
        exit()
    else:
        print("Choose correct choice")
```

OUTPUT :



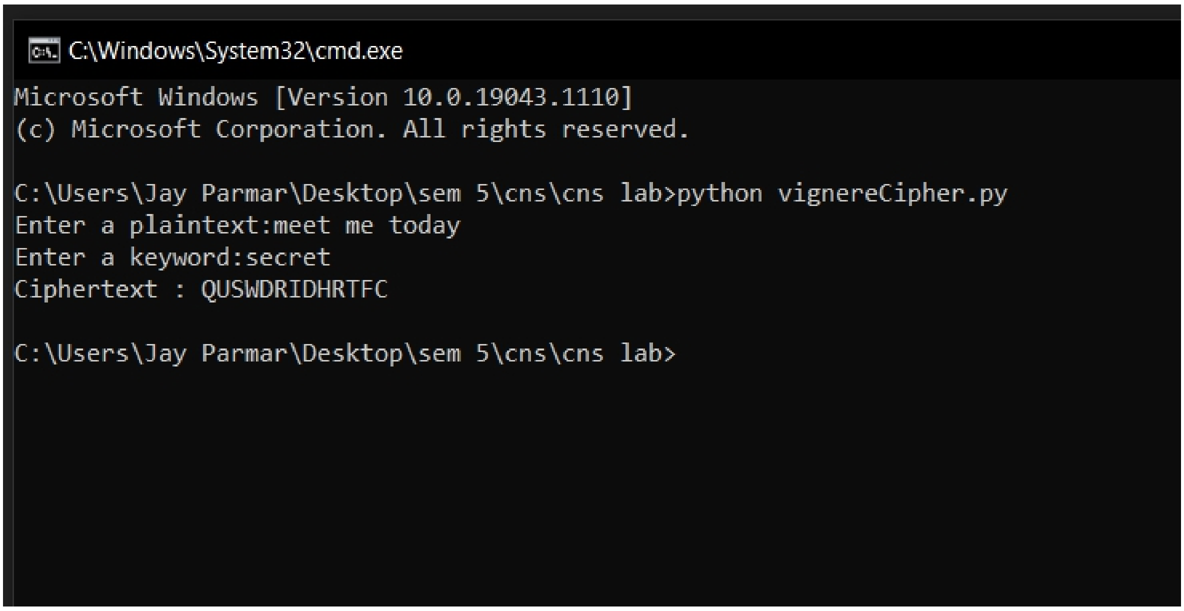
```
Microsoft Windows [Version 10.0.19043.1110]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Jay Parmar\Desktop\sem 5\cns\cns lab>python playFair.py
Enter key:platinum

 1.Encryption
 2.Decryption:
 3.Exit
Enter your choice:1
Enter plaintext:keep it safe
Cipher Text: OD DL PI QI GF
 1.Encryption
 2.Decryption:
 3.Exit
Enter your choice:
```

VIGNERE CIPHER

PROGRAM :

```
def generateKey(string, key):
    key = list(key)
    if len(string) == len(key):
        return(key)
    else:
        for i in range(len(string) -
                       len(key)):
            key.append(key[i % len(key)])
    return("". join(key))

def cipherText(string, key):
    cipher_text = []
    for i in range(len(string)):
        x = (ord(string[i]) +
             ord(key[i])) % 26
        x += ord('A')
        cipher_text.append(chr(x))
    return("" . join(cipher_text))



if __name__ == "__main__":
    string = input("Enter a plaintext:")
    keyword = input("Enter a keyword:")
    key = generateKey(string, keyword)
    cipher_text = cipherText(string,key)
    print("Ciphertext :", cipher_text)
```

OUTPUT :



```
Microsoft Windows [Version 10.0.19043.1110]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Jay Parmar\Desktop\sem 5\cns\cns lab>python vignereCipher.py
Enter a plaintext:meet me today
Enter a keyword:secret
Ciphertext : QUSWDRIDHRTFC

C:\Users\Jay Parmar\Desktop\sem 5\cns\cns lab>
```

CONCLUSION : Hence we have learned and implemented cryptanalysis or decoding playfair, vignere cipher using python.