Cryptography

Lab-I

1. WAP to encrypt and decrypt the text in Ceaser Cipher.

```
def encryption(key,plaintext):
    plaintext=plaintext.lower()
    cipherText=''
    for i in range(len(plaintext)):
        if plaintext[i]>='a' and plaintext[i]<='z':
            base='a'
        if plaintext[i] ==' ':
            cipherText=cipherText+plaintext[i]
        else:
            encrypt= chr((ord(plaintext[i])-ord(base)+key)%26 +ord(base));
            cipherText=cipherText+ (encrypt)
    return cipherText

def decryption(key,cipherText):
    cipherText=cipherText.lower(
    plainText=''
    for i in range(len(cipherText)):
        if cipherText[i]>='a' and cipherText[i]<='z':
            base='a'
        if cipherText[i] ==' ':
            plainText=plainText+cipherText[i]
        else:
            encrypt= chr((ord(cipherText[i])-ord(base)-key)%26 +ord(base));
```

```
            plainText= plainText+ (encrypt)


    return plainText;


user = str(input("Enter the plain text: "))
cipherText1=encryption(3,user)
print(f"The encrypted text is {cipherText1}")
plaintText1=decryption(3,cipherText1)
print(f"The decrypted text after encryption  is {plaintText1}")
```

2. Wap to encrypt and decrypt in Monoalphabetic Cipher.

```
def encryption(code,plaintext):
    plaintext=plaintext.lower()
    cipherText=''
    for i in range(len(plaintext)):
        if plaintext[i] ==' '
            cipherText=cipherText+plaintext[i]
        else:
            cipherText+=code[plaintext[i]]
    return cipherText


def decryption(code,cipherText):
    cipherText=cipherText.lower()
    plainText=''
    for i in range(len(cipherText)):
        if cipherText[i] ==' ':
```

```
            plainText=plainText+cipherText[i]
        else:
            plainText=plainText+get_key(cipherText[i])
    return plainText;

def get_key(val): # used to get key using value from dict
    for key, value in code.items():
        if val == value:
            return key

normalChar = "abcdefghijklmnopqrstuvwxyz"
codedChar = "qwertyuiopasdfghjklzxcvbnm"
normalChar = list(normalChar)
codedChar=list(codedChar)

code={} # dict of normalChar and codedChar
for i in range(26):
    code[normalChar[i]]=codedChar[i]

plaintext=str(input("Enter the text: "))
cipherText1=encryption(code,plaintext)
print(f"The text after encrypion is {cipherText1}")
plainText1=decryption(code,cipherText1)
print(f"The text after decrypion is {plainText1}")
```

3. WAP to encrypt and decrypt text in Playfair Cipher.

```
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 MSG:"))
    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][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


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