

Evaluation Sheet

Class: T.E Computer Engineering

Sem: VI

Subject: Cryptography and System Security

Experiment No: 2

Date: 10/01/2023

Title of Experiment: Design and Implementation of Play fair Cipher.

Sr. No.	Evaluation Criteria	Max Marks	Marks Obtained
1	Practical Performance	12	
2	Oral	2	
3	Timely Submission	1	
	Total	15	

Signature of Subject Teacher
[Vijesh M.Nair]

Program Code –

```
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 message: "))
    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("Ciphertext:",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("\nEnter ciphertext:"))

    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

encrypt()

decrypt()

```

Output –

```
Enter key: kid
Enter message: niraj
Ciphertext: MD TI DW
Enter ciphertext:MDTIDW
Plain Text: NI RA IX
```