

**Vivekanand Education Society's Institute of Technology**  
**Department of Computer Engineering**



**Subject: AI**

**Class :- CMPN**

**Semester:-6**

**Div :- D12A**

Roll No: 08	Name: Varnit Batheja		
Exp No: 09	Title: Implementation of playfair cipher algorithm.		
DOP:	25-03-2022	DOS:	1-04-2022
GRADE:		LAB OUTCOMES:	SIGNATURE:

## EXPERIMENT No 9

### AIM

Implementation of playfair cipher.

### THEORY

It is polyalphabetic cipher

Secret key arranged in a  $5 \times 5$  matrix

Sequence key of 25 letter with combined i and j.

### Encryption rules

Remove any punctuation or char that are not in key square

Identify double letters in plaintext replace occurrence with x

Break plaintext in pairs

Locate letters in key square

If letters in different rows, column, replace pair with letters on same row but at other pair of corners of rectangle

If letters appear on same row replace with letters on right

If letters appear on same column, replace with letter below

### CONCLUSION

In this experiment we studied and understood and implemented program on playfair cipher successfully.

Teacher's Sign

```

key=input("Enter key\n")
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:\n",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:\n",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\n"))
    if choice==1:
        encrypt()
    elif choice==2:
        decrypt()
    elif choice==3:
        exit()
    else:
        print("Choose correct choice")

```

```

Enter key
11573

1.Encryption
2.Decryption:
3.EXIT
1
ENTER MSG:Muskan M H
CIPHER TEXT:
PR UH 5Q NG
1.Encryption
2.Decryption:
3.EXIT
|

```