

Exp: 1B**Playfair Cipher****Date: 03-02-2024****Aim:**

To write a python program implementing playfair cipher algorithm

Algorithm:

1. Get the plaintext from the user
2. Get the key from the user
3. Plaintext is encrypted two letters at a time
4. If a pair is a repeated letter, insert filler like 'X'
5. If both letters fall in the same row, replace each with letter to right (wrapping back to start from end)
6. If both letters fall in the same column, replace each with the letter below it (again wrapping to top from bottom)
7. Otherwise each letter is replaced by the letter in the same row and in the column of the other letter of the pair.

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:
    if c not in result:
        if c=='J':
            result.append('I')
        else:
            result.append(c)
flag=0
for i in range(65,91):
    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)
for i in range(0,5): 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():
    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("{}
        {}".form
        at(my_
        matrix[l
        oc[0]][l
        oc1[1]],
        my_ma
        trix[loc1
        [0]][loc[

```

```

        1]],end
        ='
    ')
    i=i+2

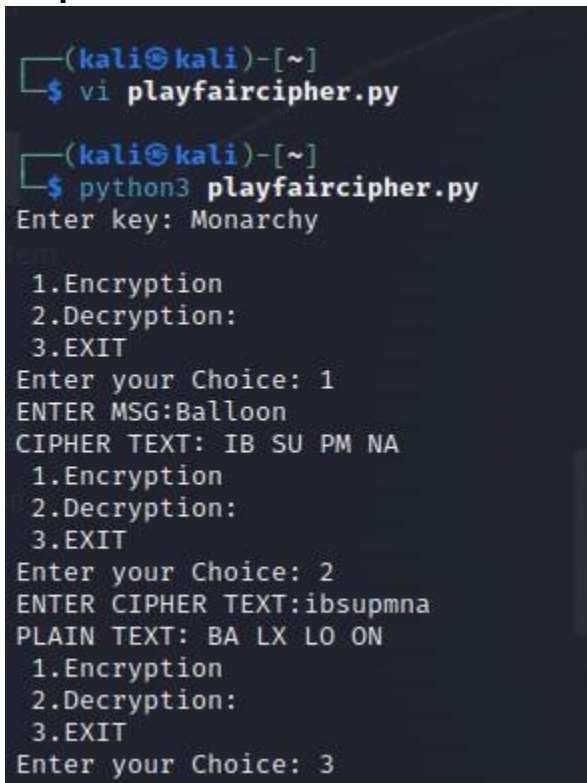
```

```

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

```

Output:



```

(kali@kali)-[~]
$ vi playfaircipher.py

(kali@kali)-[~]
$ python3 playfaircipher.py
Enter key: Monarchy

1.Encryption
2.Decryption:
3.EXIT
Enter your Choice: 1
ENTER MSG:Balloon
CIPHER TEXT: IB SU PM NA
1.Encryption
2.Decryption:
3.EXIT
Enter your Choice: 2
ENTER CIPHER TEXT:ibsupmna
PLAIN TEXT: BA LX LO ON
1.Encryption
2.Decryption:
3.EXIT
Enter your Choice: 3

```

Result: Thus the python program for playfair cipher is implemented successfully.