**Practical No : 03**

**Problem Statement:**

Write a program to implement playfair cipher

**Program:**

import sys

def find\_position(matrix, target):

for i in range(len(matrix)):

for j in range(len(matrix[i])):

if matrix[i][j] == target:

return i, j

return None, None

def createMatrix(keyword):

matrix = [["" for \_ in range(5)] for \_ in range(5)]

l1 = []

len1,len2 = 0,0

row,col=0,0

for x in range(97, 123):

if chr(x) not in keyword and chr(x) not in 'j':

l1.append(chr(x))

for i in range(5):

for j in range(5):

col = col % 5

if (len1 < len(keyword)):

matrix[row][col] = keyword[len1]

len1 += 1

else:

matrix[row][col] = l1[len2]

len2 += 1

col += 1

row += 1

return matrix

def playfair\_encryption(plain\_text, mat):

ciphertext = ""

i=0

while i < len(plain\_text):

t1 = plain\_text[i]

t2 = plain\_text[i+1]

r1,c1 = find\_position(mat, t1)

r2,c2 = find\_position(mat, t2)

if c1 == c2:

ciphertext += mat[(r1+1) % 5][c1]

ciphertext += mat[(r2+1) % 5][c2]

elif r1 == r2:

ciphertext += mat[r1][ (c1+1) % 5]

ciphertext += mat[r2][ (c2+1) % 5]

else:

ciphertext += mat[r1][c2]

ciphertext += mat[r2][c1]

i += 2

return ciphertext

def encryption(f, mat):

plain\_text = f.read()

p\_text = ""

cipher\_text = ""

i=0

while(i < len(plain\_text)):

if i == len(plain\_text)-1:

p\_text += plain\_text[i]

if len(p\_text)%2 != 0:

p\_text += "x"

break

elif plain\_text[i] == plain\_text[i+1]:

p\_text += plain\_text[i]

p\_text += 'x'

i += 1

else:

p\_text += plain\_text[i]

p\_text += plain\_text[i+1]

i += 2

cipher\_text = playfair\_encryption(p\_text, mat)

print("-"\*(len(plain\_text)+17))

print(f"Plain Text: {plain\_text}")

print(f"Pairing Text: {p\_text}")

print(f"Cipher Text: {cipher\_text}")

print("-"\*(len(plain\_text)+17))

with open("encryption\_cipher.txt","w") as f1:

f1.write(cipher\_text)

def playfair\_decryption(c\_text, mat):

plaintext = ""

i=0

while i < len(c\_text):

t1 = c\_text[i]

t2 = c\_text[i+1]

r1,c1 = find\_position(mat, t1)

r2,c2 = find\_position(mat, t2)

if c1 == c2:

plaintext += mat[(r1-1) % 5][c1]

plaintext += mat[(r2-1) % 5][c2]

elif r1 == r2:

plaintext += mat[r1][ (c1-1) % 5]

plaintext += mat[r2][ (c2-1) % 5]

else:

plaintext += mat[r1][c2]

plaintext += mat[r2][c1]

i += 2

p\_text=""

for i in plaintext:

if i != "x":

p\_text += i

return p\_text

def decryption(f, mat):

cipher\_text = f.read()

plain\_text = playfair\_decryption(cipher\_text, mat)

print("-"\*(len(plain\_text)+17))

print(f"Cipher Text: {cipher\_text}")

print(f"Plain Text: {plain\_text}")

print("-"\*(len(plain\_text)+17))

with open("decryption\_plain.txt","w") as f1:

f1.write(plain\_text)

def user\_menu():

print("[1]. Encryption of plain text")

print("[2]. Decryption of cipher text")

print("[0]. To Terminate Program ")

choice = int(input("Enter Your Choice: "))

return choice

def main():

keyword = "abc"

res=[]

[res.append(x) for x in keyword if x not in res]

mat = createMatrix(res)

print("-"\*30)

print("\tMATRIX")

print("-"\*30)

for i in range(len(mat)):

print(mat[i])

while True:

choice = user\_menu()

if choice == 1:

with open("encryption\_plain.txt","r") as f:

encryption(f, mat)

elif choice == 2:

with open("decryption\_cipher.txt","r") as f:

decryption(f, mat)

elif choice == 0:

print("Program Terminated...")

sys.exit()

else:

print("Invalid Choice")

continue

if \_\_name\_\_ == "\_\_main\_\_":

main()

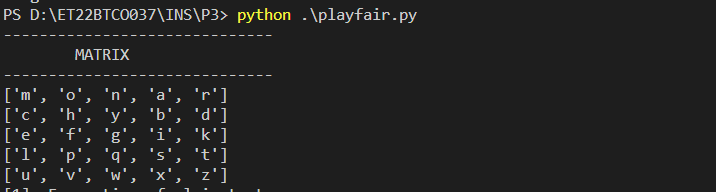
**Input:**

| **encryption\_plain.txt**  instrumentsz  hellloo | **decryption\_cipher.txt**  gatlmzclrqtx  kcnvnvmpny |
| --- | --- |

**Output:**

| **decryption\_cipher.txt**  gatlmzclrqtx  kcnvnvmpny | **decryption\_plain.txt**  instrumentsz  hellloo |
| --- | --- |

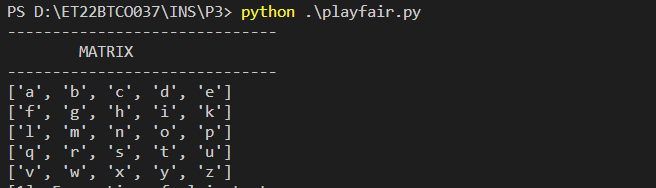
**Keyword: monarchy**

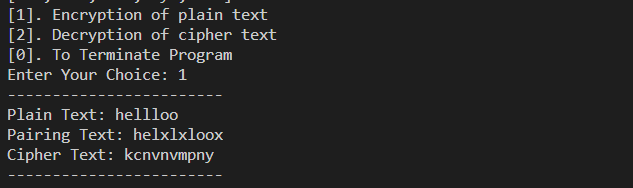
****





**Keyword: abc**

****

****

****