Name: Jigar Siddhpura **SAPID:** 60004200155

DIV: C/C2 **Branch:** Computer Engineering

IS - Experiment 1 - PLAYFAIR CIPHER

Jigar Siddhpura
60004210155
C22

Experiment 1 – Playfair cipher

Aim: To implement playfair cipher

Theory:

Playfair cipher is a symmetric cryptographic technique used to encrypt plaintext into ciphertext. A key is chosen for encryption forms a 5x5 grid where unique letters of key are arranged excludes duplicate & treating 'I' & 'J' as same letter. Once grid is made plaintext is prepared by ~~preparing~~ breaking into pairs of letters. If repeated letters are there 'X' is inserted in between. Substitution is done for this pairs:

1. If letters fall in same row replace them with immediate right of same row.
2. If letters fall in same column replace with immediate below.
3. If letters form rectangle, replace with letters on same row but other corners of rectangle.

This will give ciphertext. Decryption follows same way just reversing substitutions & gets orignal plaintext.

Eg: Key = 'PLAYFAIREXAMPLE'
Text = 'HELLOALL'
Matrix:

P	L	A	Y	F
I/J	R	E	X	M
B	C	D	G	H
K	N	O	Q	S
T	U	V	W	Z

Pair of letters : HE LX LX OA LX LX

So cipher text : DM YR YR VB YR YR

Now, decryption,

decrypted text : HE LX LX OA LX LX

Conclusion :

Hence studied & implemented play cipher code in python & tested with example.

CODE

```
import string
key = input("Enter key : ").upper()

def make_matrix(key):
    letters = list(string.ascii_uppercase)
    for letter in key:
        if letter == 'I' or letter == 'J':
            letters.remove('I')
            letters.remove('J')
        else:
            letters.remove(letter)
    matrix = list(key)
    matrix.extend(letters)
    matrix = [matrix[i:i+5] for i in range(0, 25, 5)]
    return matrix

word = input("Enter plaintext : ").upper()
print(f"key: {key}\nplain text: {word}\n")
print("matrix : ")
cipher_matrix = make_matrix(key)
for row in cipher_matrix:
    print(row)

def create_pairs(word):
    pairs = []
    if len(word) % 2 != 0:
        word += word[-1]
    for i in range(0, len(word), 2):
        pair = word[i:i+2]
        pairs.append(pair)
    return pairs

def add_extra_character(word):
    result = ""
    extra_char = find_missing_letters(word)
    for i in range(len(word) - 1):
        result += word[i]
        if word[i] == word[i + 1]:
            while extra_char in result or extra_char == word[i]:
                extra_char = chr(ord(extra_char) + 1)
            result += extra_char
    result += word[-1]
    if len(result) % 2 != 0:
```

```
        result += extra_char
    return result

def find_missing_letters(word):
    all_letters = set("abcdefghijklmnopqrstuvwxyz")
    word_letters = set(word.lower())
    missing_letters = all_letters - word_letters
    return sorted(list(missing_letters))[0]

result_word = add_extra_character(word)
result = create_pairs(result_word)
print(f"2pair of text : {result}")
mod_text = result

def search(matrix, elmt):
    for i in range(5):
        for j in range(5):
            if matrix[i][j] == elmt:
                return i, j

def playfair_encrypt(mat, txt):
    cipher = []
    for i in txt:
        a, b = search(mat, i[0])
        c, d = search(mat, i[1])
        if b == d:
            word = ""
            if a + 1 > 4:
                word += mat[0][b]
            else:
                word += mat[a + 1][b]
            if c + 1 > 4:
                word += mat[0][d]
            else:
                word += mat[c + 1][d]
            cipher.append(word)
        elif a == c:
            word = ""
            if b + 1 > 4:
                word += mat[a][0]
            else:
                word += mat[a][b + 1]
            if d + 1 > 4:
                word += mat[c][0]
            else:
                word += mat[c][d + 1]
            cipher.append(word)
```

```
        else:
            word = mat[a][d] + mat[c][b]
            cipher.append(word)
    return cipher

def playfair_decrypt(mat, txt):
    plaintext = []
    for i in txt:
        a, b = search(mat, i[0])
        c, d = search(mat, i[1])
        if b == d:
            word = ""
            if a - 1 < 0:
                word += mat[4][b]
            else:
                word += mat[a - 1][b]
            if c - 1 < 0:
                word += mat[4][d]
            else:
                word += mat[c - 1][d]
            plaintext.append(word)
        elif a == c:
            word = ""
            if b - 1 < 0:
                word += mat[a][4]
            else:
                word += mat[a][b - 1]
            if d - 1 < 0:
                word += mat[c][4]
            else:
                word += mat[c][d - 1]
            plaintext.append(word)
        else:
            word = mat[a][d] + mat[c][b]
            plaintext.append(word)
    return plaintext

cipher_txt = "".join(playfair_encrypt(cipher_matrix, mod_text))
print(f"Cipher Text is : {cipher_txt}")
decrypt_txt = "".join(playfair_decrypt(cipher_matrix, create_pairs(cipher_txt)))
print(f"Decrypted Text : {decrypt_txt}")
```

OUTPUT

```
PS D:\SEM-6\IS\EXPERIMENTS> python -u "d:\SEM-6\IS\EXPERIMENTS\playfair.py"
Enter key : cipher
Enter plaintext : cryptography
key: CIPHER
plain text: CRYPTOGRAPHY

matrix :
['C', 'I', 'P', 'H', 'E']
['R', 'A', 'B', 'D', 'F']
['G', 'K', 'L', 'M', 'N']
['O', 'Q', 'S', 'T', 'U']
['V', 'W', 'X', 'Y', 'Z']
2pair of text : ['CR', 'YP', 'TO', 'GR', 'AP', 'HY']
Cipher Text is : RGXHUQOGBIDH
Decrypted Text : CRYPTOGRAPHY
PS D:\SEM-6\IS\EXPERIMENTS>
```