**Ex. No.:1B**

**Date: 03/02/2024**

**PLAYFAIR CIPHER**

**Aim:**

To write a Python program to encrypt and decrypt a Plain Text using Playfair Cipher.

**Algorithm:**

1. The Plain Text will split into pairs of two letters.
2. If there is an odd number of letters, alpha is added to the last letter.
3. A pair can’t be made as the same letter alpha. Break the letter in single and add a bogus letter to previous letter.
4. If the letter is standing alone in the process of pouring then add an extra bogus letter with a lone letter.
5. If both letters are in the same column, take the letter below each one.
6. If both the letters are on the same row, take the letter to the right of each one.

**Program:**

**#Playfair Cipher**

def playfair\_cipher(plaintext, key, mode):

# Define the alphabet, excluding 'j'

alphabet = 'abcdefghiklmnopqrstuvwxyz'

# Remove whitespace and 'j' from the key and convert to lowercase

key = key.lower().replace(' ', '').replace('j', 'i')

# Construct the key square

key\_square = ""

for letter in key + alphabet:

if letter not in key\_square:

key\_square += letter

# Split the plaintext into digraphs, padding with 'x' if necessary

plaintext = plaintext.lower().replace(' ', '').replace('j', 'i')

if len(plaintext) % 2 == 1:

plaintext += 'x'

digraphs = [plaintext[i:i+2] for i in range(0, len(plaintext), 2)]

#Define the encryption/decryption functions

def encrypt(digraph):

a,b = digraph

rowa, cola = divmod(key\_square.index(a), 5)

rowb, colb = divmod(key\_square.index(b), 5)

if rowa == rowb:

cola = (cola + 1)%5

colb = (colb + 1)%5

elif cola == colb:

rowa = (rowa + 1)%5

rowb = (rowb + 1)%5

else:

cola, colb = colb, cola

return key\_square[rowa\*5 + cola] + key\_square[rowb\*5 + colb]

def decrypt(digraph):

a,b = digraph

rowa, cola = divmod(key\_square.index(a), 5)

rowb, colb = divmod(key\_square.index(b), 5)

if rowa == rowb:

cola = (cola - 1)%5

colb = (colb - 1)%5

elif cola == colb:

rowa = (rowa - 1)%5

rowb = (rowb - 1)%5

else:

cola, colb = colb, cola

return key\_square[rowa\*5 + cola] + key\_square[rowb\*5 + colb]

# Encrypt or Decrypt the plaintext

result = ''

for digraph in digraphs:

if mode == 'encrypt':

result += encrypt(digraph)

elif mode == 'decrypt':

result += decrypt(digraph)

return result

plaintext = input("Enter Plain Text:")

key = input("Enter Key:")

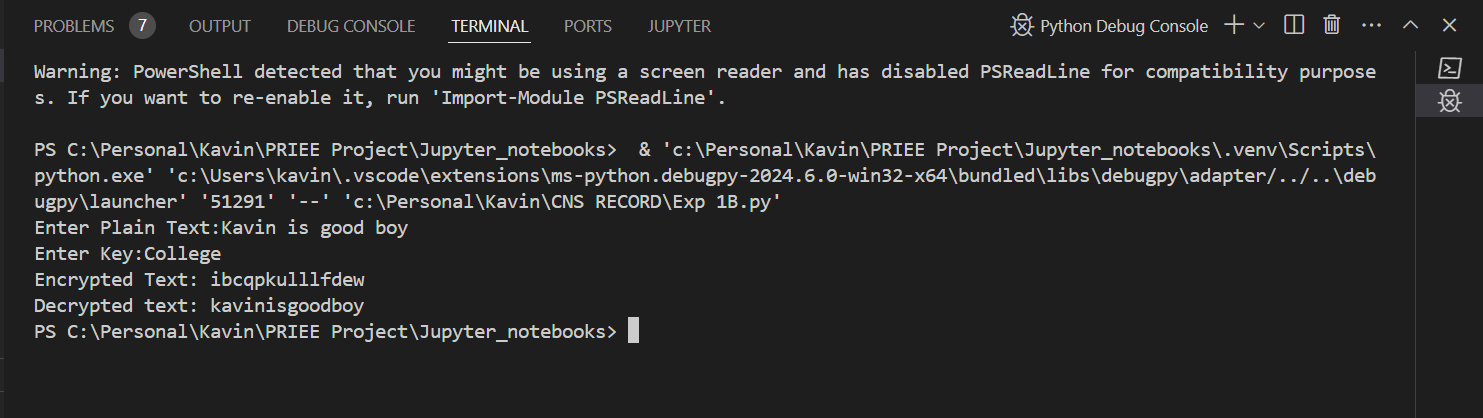
ciphertext = playfair\_cipher(plaintext, key, 'encrypt')

print(ciphertext)

decrypted\_text = playfair\_cipher(ciphertext, key, 'decrypt')

print(decrypted\_text)

**Output:**



**Result:**

Hence, the Playfair Cipher program has been implemented successfully.