

In [2]:

```

1  # Python program to implement Playfair Cipher 5*5
2  def toLowerCase(text):
3      return text.lower()
4
5  # Function to remove all spaces in a string
6
7
8  def removeSpaces(text):
9      newText = ""
10     for i in text:
11         if i == " ":
12             continue
13         else:
14             newText = newText + i
15     return newText
16
17 # Function to group 2 elements of a string
18 # as a list element
19
20
21 def Diagraph(text):
22     Diagraph = []
23     group = 0
24     for i in range(2, len(text), 2):
25         Diagraph.append(text[group:i])
26
27         group = i
28     Diagraph.append(text[group:])
29     return Diagraph
30
31 # Function to fill a letter in a string element
32 # If 2 letters in the same string matches
33
34
35 def FillerLetter(text):
36     k = len(text)
37     if k % 2 == 0:
38         for i in range(0, k, 2):
39             if text[i] == text[i+1]:
40                 new_word = text[0:i+1] + str('x') + text[i+1:]
41                 new_word = FillerLetter(new_word)
42                 break
43             else:
44                 new_word = text
45     else:
46         for i in range(0, k-1, 2):
47             if text[i] == text[i+1]:
48                 new_word = text[0:i+1] + str('x') + text[i+1:]
49                 new_word = FillerLetter(new_word)
50                 break
51             else:
52                 new_word = text
53     return new_word
54
55
56 list1 = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'k', 'l', 'm',
57          'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']

```

```
58
59 # Function to generate the 5x5 key square matrix
60
61
62 def generateKeyTable(word, list1):
63     key_letters = []
64     for i in word:
65         if i not in key_letters:
66             key_letters.append(i)
67
68     compElements = []
69     for i in key_letters:
70         if i not in compElements:
71             compElements.append(i)
72     for i in list1:
73         if i not in compElements:
74             compElements.append(i)
75
76     matrix = []
77     while compElements != []:
78         matrix.append(compElements[:5])
79         compElements = compElements[5:]
80
81     return matrix
82
83
84 def search(mat, element):
85     for i in range(5):
86         for j in range(5):
87             if(mat[i][j] == element):
88                 return i, j
89
90
91 def encrypt_RowRule(matr, e1r, e1c, e2r, e2c):
92     char1 = ''
93     if e1c == 4:
94         char1 = matr[e1r][0]
95     else:
96         char1 = matr[e1r][e1c+1]
97
98     char2 = ''
99     if e2c == 4:
100         char2 = matr[e2r][0]
101     else:
102         char2 = matr[e2r][e2c+1]
103
104     return char1, char2
105
106
107 def encrypt_ColumnRule(matr, e1r, e1c, e2r, e2c):
108     char1 = ''
109     if e1r == 4:
110         char1 = matr[0][e1c]
111     else:
112         char1 = matr[e1r+1][e1c]
113
114     char2 = ''
```

```

115     if e2r == 4:
116         char2 = matr[0][e2c]
117     else:
118         char2 = matr[e2r+1][e2c]
119
120     return char1, char2
121
122
123 def encrypt_RectangleRule(matr, e1r, e1c, e2r, e2c):
124     char1 = ''
125     char1 = matr[e1r][e2c]
126
127     char2 = ''
128     char2 = matr[e2r][e1c]
129
130     return char1, char2
131
132
133 def encryptByPlayfairCipher(Matrix, plainList):
134     CipherText = []
135     for i in range(0, len(plainList)):
136         c1 = 0
137         c2 = 0
138         ele1_x, ele1_y = search(Matrix, plainList[i][0])
139         ele2_x, ele2_y = search(Matrix, plainList[i][1])
140
141         if ele1_x == ele2_x:
142             c1, c2 = encrypt_RowRule(Matrix, ele1_x, ele1_y, ele2_x, ele2_y)
143             # Get 2 Letter cipherText
144         elif ele1_y == ele2_y:
145             c1, c2 = encrypt_ColumnRule(Matrix, ele1_x, ele1_y, ele2_x, ele2_y)
146         else:
147             c1, c2 = encrypt_RectangleRule(
148                 Matrix, ele1_x, ele1_y, ele2_x, ele2_y)
149
150         cipher = c1 + c2
151         CipherText.append(cipher)
152     return CipherText
153
154
155 #text_Plain = 'instruments'
156 text_Plain = input("Enter the Text_Plain :- ")
157 text_Plain = removeSpaces(toLowerCase(text_Plain))
158 PlainTextList = Diagraph(FillerLetter(text_Plain))
159 if len(PlainTextList[-1]) != 2:
160     PlainTextList[-1] = PlainTextList[-1]+'z'
161
162 #key = "Monarchy"
163 #print("Key text:", key)
164 key = input("Enter the Key:- ")
165 key = toLowerCase(key)
166 Matrix = generateKeyTable(key, list1)
167
168 print("Plain Text:", text_Plain)
169 CipherList = encryptByPlayfairCipher(Matrix, PlainTextList)
170
171 CipherText = ""

```

```
172 for i in CipherList:
173     CipherText += i
174 print("CipherText:", CipherText)
175
176 # This code is Contributed by Boda_Venkata_Nikith
177
```

Enter the Text_Plain :- Amar Deep

Enter the Key:- Amar

Plain Text: amardeep

CipherText: mrmbefdq

In []:

1