

Play Fair Cipher

Code:

```

1  def split(message):
2      msg = []
3      len_msg = len(message)
4      counter = 0
5      while (True):
6          split = []
7          if counter >= len_msg:
8              break
9          if (counter == len_msg-1):
10             split.append(message[counter])
11             split.append("z")
12             msg.append(split)
13             break
14             if (message[counter] == message[counter + 1]):
15                 split.append(message[counter])
16                 split.append("x")
17                 msg.append(split)
18                 counter += 1
19             else:
20                 split.append(message[counter])
21                 split.append(message[counter+1])
22                 msg.append(split)
23                 counter += 2
24         return msg
25
26 def sqRule(a_col, b_col, a_row, b_row, table):
27     return (table[a_row][b_col], table[b_row][a_col])
28
29 def colRule(a_col, b_col, a_row, b_row, table):
30     return (table[(a_row+1)%5][a_col], table[(b_row+1)%5][b_col])
31
32 def rowRule(a_col, b_col, a_row, b_row, table):
33     return (table[a_row][(a_col+1)%5], table[b_row][(b_col+1)%5])
34
35 def find_letter_index(table, letter):
36     for i in range(len(table)):
37         for j in range(len(table[i])):
38             if table[i][j] == letter:
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
38         if table[i][j] == letter:
39             return i, j
40
41 def encrypt(table, mes):
42     result = []
43     for i in range(len(mes)):
44         a_row, a_col = find_letter_index(table, mes[i][0])
45         b_row, b_col = find_letter_index(table, mes[i][1])
46
47         if a_row == b_row:
48             result.append(rowRule(a_col, b_col, a_row, b_row, table))
49         elif a_col == b_col:
50             result.append(colRule(a_col, b_col, a_row, b_row, table))
51         else:
52             result.append(sqRule(a_col, b_col, a_row, b_row, table))
53     print(result)
54     return result
55
56 def DEcolRule(a_col, b_col, a_row, b_row, table):
57     return (table[(a_row-1)%5][a_col], table[(b_row-1)%5][b_col])
58
59 def DErowRule(a_col, b_col, a_row, b_row, table):
60     return (table[a_row][(a_col-1)%5], table[b_row][(b_col-1)%5])
61
62 def decrypt(table, mes):
63     result = []
64     for i in range(len(mes)):
65         a_row, a_col = find_letter_index(table, mes[i][0])
66         b_row, b_col = find_letter_index(table, mes[i][1])
67
68         if a_row == b_row:
69             result.append(DErowRule(a_col, b_col, a_row, b_row, table))
70         elif a_col == b_col:
71             result.append(DEcolRule(a_col, b_col, a_row, b_row, table))
72         else:
73             result.append(sqRule(a_col, b_col, a_row, b_row, table))
74     print(result)
75     return result
76
77 def table(key):
78     alphabets = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
79     key_length = len(key)
80     counter = 0
81     matrix = []
82     for i in range(5):
83         row = []
84         for j in range(5):
85             if counter < key_length:
86                 row.append(key[counter])
87                 alphabets.remove(key[counter])
88                 counter += 1
89             else:
90                 row.append(alphabets.pop(0))
91         matrix.append(row)
92     return matrix
93
94 key = input("Enter key : ")
95 message = input("Enter Message : ")
96
97 t = table(key)
98 for row in t:
99     print(row)
100
101 mes = split(message)
102 print(mes)
103
104 choice = int(input("1. Encrypt\n2. Decrypt\n"))
105
106 if (choice == 1):
107     encrypt(t, mes)
108 if (choice == 2):
109     decrypt(t, mes)

```

Output:

```
Enter key : monarchy
Enter Message : secret
['m', 'o', 'n', 'a', 'r']
['c', 'h', 'y', 'b', 'd']
['e', 'f', 'g', 'i', 'k']
['l', 'p', 'q', 's', 't']
['u', 'v', 'w', 'x', 'z']
[['s', 'e'], ['c', 'r'], ['e', 't']]
1. Encrypt
2. Decrypt
1
[('l', 'i'), ('d', 'm'), ('k', 'l')]

Enter key : monarchy
Enter Message : lidmkl
['m', 'o', 'n', 'a', 'r']
['c', 'h', 'y', 'b', 'd']
['e', 'f', 'g', 'i', 'k']
['l', 'p', 'q', 's', 't']
['u', 'v', 'w', 'x', 'z']
[['l', 'i'], ['d', 'm'], ['k', 'l']]
1. Encrypt
2. Decrypt
2
[('s', 'e'), ('c', 'r'), ('e', 't')]
```