Name: Muhammad Rafiq

Seat No: B17101061

Course: Network Security & Cryptography (NS'21 Lab)

Assignment : Assignment # 6

Section: A

Task 1: Write a program for Play Fiar Cipher that can encrypt and decrypt text it..

PlayFair Cipher

In [1]:

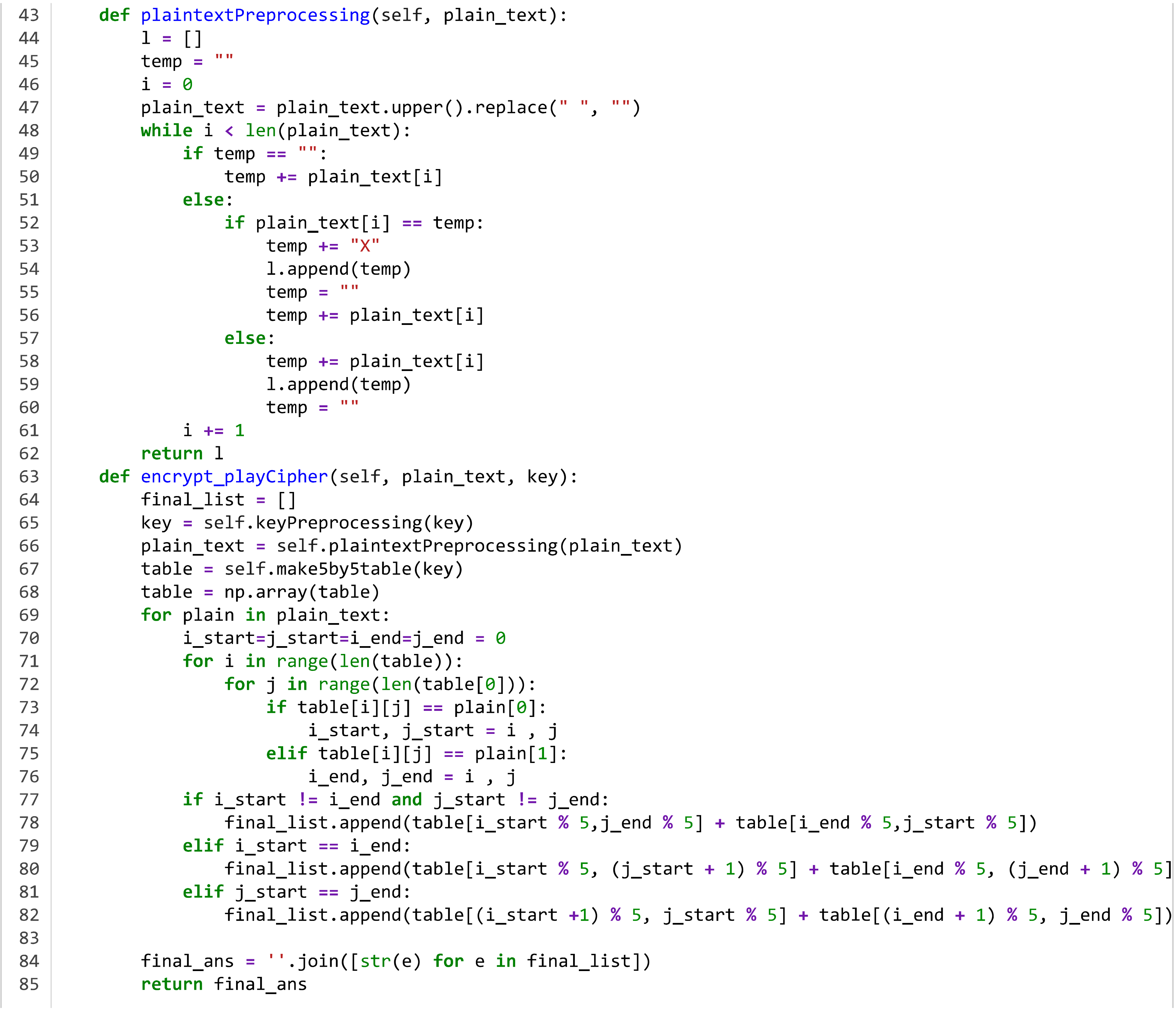
```python
import numpy as np
class PlayFairCipher:

    def __init(self):
        print("initialized")

    def make5by5table(self, key):
        index =0
        alphabet = 65
        l = [[0]*5 for i in range(5)]
        for i in range(len(l)):
            for j in range(len(l[0])):
                if index < len(key):
                    l[i][j] = key[index]
                    index += 1
                else:
                    flag = True
                    while(flag):
                        if chr(alphabet) not in key+"J":
                            l[i][j] = chr(alphabet)
                            alphabet += 1
                            flag = False
                        else:
                            alphabet += 1
        return l


    def keyPreprocessing(self, key):
        key = key.upper()
        unique_dicts = {}
        for i in key:
            if i not in unique_dicts:
                unique_dicts[i] = 1

        key ="".join([str(e) for e in unique_dicts.keys()])
        return key
```



```python
    def plaintextPreprocessing(self, plain_text):
        l = []
        temp = ""
        i = 0
        plain_text = plain_text.upper().replace(" ", "")
        while i < len(plain_text):
            if temp == "":
                temp += plain_text[i]
            else:
                if plain_text[i] == temp:
                    temp += "X"
                    l.append(temp)
                    temp = ""
                    temp += plain_text[i]
                else:
                    temp += plain_text[i]
                    l.append(temp)
                    temp = ""
            i += 1
        return l
    def encrypt_playCipher(self, plain_text, key):
        final_list = []
        key = self.keyPreprocessing(key)
        plain_text = self.plaintextPreprocessing(plain_text)
        table = self.make5by5table(key)
        table = np.array(table)
        for plain in plain_text:
            i_start=j_start=i_end=j_end = 0
            for i in range(len(table)):
                for j in range(len(table[0])):
                    if table[i][j] == plain[0]:
                        i_start, j_start = i , j
                    elif table[i][j] == plain[1]:
                        i_end, j_end = i , j
            if i_start != i_end and j_start != j_end:
                final_list.append(table[i_start % 5,j_end % 5] + table[i_end % 5,j_start % 5])
            elif i_start == i_end:
                final_list.append(table[i_start % 5, (j_start + 1) % 5] + table[i_end % 5, (j_end + 1) % 5]
            elif j_start == j_end:
                final_list.append(table[(i_start +1) % 5, j_start % 5] + table[(i_end + 1) % 5, j_end % 5])

        final_ans = ''.join([str(e) for e in final_list])
        return final_ans

    def decrypt_playCipher(self, cipher_text, key):
        final_list = []
        key = self.keyPreprocessing(key)
        cipher_text = self.plaintextPreprocessing(cipher_text)
        table = self.make5by5table(key)
        table = np.array(table)
        for plain in cipher_text:
            i_start=j_start=i_end=j_end = 0
            for i in range(len(table)):
                for j in range(len(table[0])):
                    if table[i][j] == plain[0]:
                        i_start, j_start = i , j
                    elif table[i][j] == plain[1]:
                        i_end, j_end = i , j
            if i_start != i_end and j_start != j_end:
                final_list.append(table[i_start % 5,j_end % 5] + table[i_end % 5,j_start % 5])
            elif i_start == i_end:
                final_list.append(table[i_start % 5, (j_start - 1) % 5] + table[i_end % 5, (j_end - 1) % 5]
            elif j_start == j_end:
                final_list.append(table[(i_start - 1) % 5, j_start % 5] + table[(i_end - 1) % 5, j_end % 5]

        final_ans = ''.join([str(e) for e in final_list])
        return final_ans

```

Task 2: Encrypt the plain text "Palyfair Cipher is completely different way to encrypt the message", where key is "Success".

In [2]:

```python
play = PlayFairCipher()
plain_text= "Playfair Cipher is completely different way to encrypt the message"
cipher_text = play.encrypt_playCipher(plain_text, "success")

print("Text After encryption of a plain text given in a question would become: \n")
print(cipher_text)
```

```
Text After encryption of a plain text given in a question would become: 

QKEZHCMOSLTDGYOBSQKRMCRAMXBKLCGCYGTZZUZRRSLAYEQOZNGRAUUSMG
```

In [3]:

```python
plain_text = play.decrypt_playCipher(cipher_text, "success")

print("Text after decryption of cipher text which we encrypted before would become: \n")
print(plain_text)
```

```
Text after decryption of cipher text which we encrypted before would become: 

PLAYFAIRCIPHERISCOMPLETELYDIFXFERENTWAYTOENCRYPTTHEMESSAGE
```

----------END----------

Muhammad Rafiq
BA101061

Assignment. Play fair Cipher.

Encrypt "Playfair cipher is completely different way to encrypt the message"
key is "Success"

Generating table of 5×5

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

Generating Pairs.

PL AY FA IR CI PH ER IS CO
MP LE TE LY DI FX FE RE
NT WA YT OE NC RY PT TH
EM ES SA GE

Encrypted message would becomes
with key = Success.

QKEZHCMO SLTD GY OBS QKRMC RA
MXBKL CGCYG TZZUZ RRS LAYE
QOZNGRAUUSMG.