HW2 - DES

這次使用的語言為 Python，實作方式為，首先將輸入字串以 8 個字元為單位分割放入陣列中，未滿 8 字元則塞入空白。接下來進 FOR 迴圈，FOR 迴圈疊代次數依據陣列大小而定，進入 FOR 迴圈後會先將字元轉成 ASCII，然後再轉成二進制數字。

```
for uncryedStr in data:
    BinData = text2Bin(uncryedStr)
```

接下來做 IP，由於後面寫 Code 時發現 permutation 的動作都一致，僅差在 Table 不同，因此將 permutation 獨立成一個函式

```
def initialPermutation(data : str):
    ip = [ 58, 50, 42, 34, 26, 18, 10, 2,
           60, 52, 44, 36, 28, 20, 12, 4,
           62, 54, 46, 38, 30, 22, 14, 6,
           64, 56, 48, 40, 32, 24, 16, 8,
           57, 49, 41, 33, 25, 17,  9, 1,
           59, 51, 43, 35, 27, 19, 11, 3,
           61, 53, 45, 37, 29, 21, 13, 5,
           63, 55, 47, 39, 31, 23, 15, 7  ]
    return permutation(data, ip)
```

再來將 IP 後的結果切成兩半，此時開始處理 KEY，將 KEY 讀入轉成二進制，並做 PC1，然後切割成 C、D。再來做位移，並將 C、D 合併做 PC2，以上 PC1 後的流程重複 16 次，並將結果存於陣列。

```
pc1_key = permutation(key64, pc_1)
c = pc1_key[0:28]
d = pc1_key[28:56]
pc2_keys = []
for i in range(16):
    # print('subkey ', i)
    c = shiftLeft(c, shift_num[i])
    d = shiftLeft(d, shift_num[i])
    # print('L : ')
    # print(c)
    # print('R : ')
    # print(d)
    cd_key = c + d
    pc2_key = permutation(cd_key, pc_2)
    # print('PC2 : ')
    # print(pc2_key)
    pc2_keys.append(pc2_key)
return pc2_keys
```

回到主程式

接下來就是照換位的順序，將 Subkey 與右邊資料做 F-Function

而 F-Function 就跑一堆 S-Box 將每個 S-Box 上的 6 bit 位元依 index 找到相應的值並轉成 2 進制回傳(4 bit)

```
boxed = ''
for i in range(8):
    data6 = data48[i * 6 : (i + 1) * 6]
    row_index = int(data6[0] + data6[-1], 2)
    col_index = int(data6[1:5], 2)
    value = s_box[i][row_index][col_index]
    boxed += dec2Bin(value, 4)
return boxed
```

F-function 的結果與左邊資料做 XOR，再來重複 16 次，最後左右合併，再將結果做 FP，存入一個字串，結束此次 LOOP，最後所有的字元都處理完後，將此字串轉成 16 進制 Print 出來即完成。

```
    for i in range(16):
        pre_r_block = r_block
        r_block = xor(l_block, feistel(pre_r_block, keys[i]))
        l_block = pre_r_block
    whole_block = r_block + l_block
    cryed += fianlPermutation(whole_block)
# print(len(cryed), cryed)
print(bin2Hex(cryed))
```