Quiz 5

309552026

鄭偉丞

1. $1 = C_7 + C_6 + C_5 + C_4 + C_3 + C_2 + C_1$

   $1 = C_7 + C_6 + C_5 + C_4 + C_3 + C_2$

   $1 = C_7 + C_6 + C_5 + C_4 + C_3$

   $1 = C_7 + C_6 + C_5 + C_4$

   $1 = C_7 + C_6 + C_5$

   $1 = C_7 + C_6$

   $1 = C_7$

   $1 = C_0$

   ➔

   $C_0 = C_7 = 1$

   $C_1 = C_2 = C_3 = C_4 = C_5 = C_6 = 0$

   ➔

   $x^8 + x + 1$

   ➔

   $a_n + a_{n+1} = a_{n+8}$
2. initial fill 8 bit seed key: 10000000

   PlanText: GODSAVEOURGRACIOUSQUEEN
3. The length is 63, not the max length which is $2^8 - 1$

code result:

```
PlainText = GODSAVEOURGRACIOUSQUEEN
maximal length = 63
```

Explain code:

```
def LFSR(key):
    nextKey = BitArray(length=8)
    for i in range(7):
        nextKey[i] = key[i] ^ key[i+1]
    nextKey[7] = key[7] ^ nextKey[0]
    return nextKey

if __name__ == "__main__":
    C = ["11000111", "11001110", "11000110", "11010100", "11001001",
        "11001111", "11101111", "10110000", "01010100", "01010001",
        "01000010", "01011101", "01010000", "01110000", "00011100",
        "10110001", "01010111", "01010101", "01011011", "01001011",
        "01100111", "00100011", "11100101"]
    seed = BitArray(bin='0b10000000')

    key = seed
    print("PlainText = ", end='')
    for i in range(len(C)):
        P = chr((BitArray(bin=C[i]) ^ key).uint)
        print(P, end='')
        key = LFSR(key)
```

透過手動分析得到 init_seed 是 10000000，將其帶入, LFSR 每次會 shift 8 bits，所

生成的 key 去跟 CipherText 做 xor, 就可以得到 PlainText.

```
# Check the max len of key cycle
keys = []
key = seed
keys.append(key.bin)
while True:
    key = LFSR(key)
    if (key.bin in keys):
        break
    else:
        keys.append(key.bin)
print("\nmaximal length =", len(keys))
```

這部分是分析可以生成多少 key,將每一個生成的 key 存成 list, 每次新的 key 都去判斷是否有重複,最終可以得到他可以生成 63 個 key.