Quiz 4

309552026
鄭偉丞

1.

```python
from bitstring import BitArray

def LFSR(tap, seed):
    xor = seed << 1
    xor[-1] = seed[0] ^ seed[seed.len - 1 - tap]
    return xor

if __name__ == '__main__':
    tap = int(input("tap= "))
    seed = BitArray(bin=input("seed= "))
    key = seed
    for i in range(10):
        key = LFSR(tap, key)
        print(key.bin, int(key[key.len - 1]))
```

先將 seed 向左 shift 存到 xor 變數中,再把 xor 變數最後一位換成題目要求的兩位的 xor,以達到跟 generator 相同的效果.

1. Bonus

```python
def Generate(tap, step, seed):
    for i in range(step):
        xor = seed << 1
        xor[-1] = seed[0] ^ seed[seed.len - 1 - tap]
        seed = xor
    return seed
```

Generate 依據 input 的 step,輸出對應步數後的 key,其中使用的 LFSR function 就是第一題中使用的.

```python
def Bit(num):
    zero = BitArray(length=8)
    target = BitArray(bin=bin(num))
    for i in range(target.len):
        zero[-i-1] = target[-i-1]
    return zero
```

由於從 int 轉 bitarray,最前面是 0 的位數會被省略而導致 array 的長度小於 8,之後位數不同會無法做 xor.所以這裡寫一個 function 來將 int 轉過來的 bitarray 變成 8 位的.

```python
if __name__ == '__main__':
    image = Image.open(input("png File name: "))

    tap = 16
    seed = BitArray(bin='0b0110100001010001000')
    key = seed
    for i in range(image.width):
        for j in range(image.height):
            RGB = list(image.getpixel((i, j)))
            for k in range(3):
                key = Generate(tap, 8, key)
                RGB[k] = (Bit(RGB[k]) ^ key[-8:]).uint
            image.putpixel((i, j), tuple(RGB))
    image.show()
```

main function 主要是從圖片依序讀出每個 pixel 的 RGB 值,在用 Generate 產出 key 並 xor 做加密/解密.

2.

```
PS C:\Users\jordan\Google 雲端硬碟\研究所\研究所上課資訊\碩一下\密碼工程\Quiz\Quiz_4> python .\Q2.py
Sequence 1: 1111000100110101111000100110101111000100110101111000100110101111000100110101111000100110101111000100
Sequence 2: 1111000100110101111000100110101111000100110101111000100110101111000100110101111000100110101111000100
```

這邊列出來兩個 generator 所產出數列的前 100 項,兩個出來的數列是一樣的

2-1. 從上圖觀察後可以得到一直在重複出現 111100010011010

2-2. $2^7 - 1$

3.

```
PS C:\Users\jordan\Google 雲端硬碟\研究所\研究所上課資訊\碩一下\密碼工程\Quiz\Quiz_4> python .\Q5.py
1000 init:0001111010110010001111010110010001111010110010001111010110010001111010110010001111010110010001111010
1111 init:1111010110010001111010110010001111010110010001111010110010001111010110010001111010110010001111010110
0110 init:0110010001111010110010001111010110010001111010110010001111010110010001111010110010001111010110010001
```

各組的循環數列分別為:

a. 000111101011001

b. 111101011001000

c. 011001000111101

各組的循環是一樣的,只是起始的位置不一樣