Part 1

1. Encrypted message of “Remember, A Jedi’s Strength Flows From The Force. But Beware: Anger, Fear, Aggression. The Dark Side, Are They.” by using a 4-bit LFSR with the seed “1001” with the first and last bit XORd with each other.

```
What is your seed? (enter 4 bits)
1001
What is your plaintext?
Remember, a Jedi's Strength Flows From The Force. But Beware: Anger, Fear, Aggression. The Dark Side, Are They.
Here is your encryption:
 0b1010010011001010110110101100101011011010110001001100101011100100010110000100000011000010010000001001010011001010110010001101
00100100111011100110010000000101001101110100011100100110010101101110011001110111010001101000001000000100011001101100011011110111
01110111001100100000001000110011100100110111101101101001000000101010001101000011001010010000000100011001101111011100100110001101
00101001011100010000000100001001110101011101000010000000100001001100101011101110110000101110010011001010011101000100000001000001
01
10111001100111011001010111001000101100001000000010001100110010101100001011100100010110000100000001000001011001110110111001101100
1
10101110100111000010111000100111010011110011011101111111110010110111011110111110001000101000110011001011001110000110110011011001
00011101001101110001001110101100100100001011110111001011011000101010010000100010100011001101101100110001011110111100101100010 01
1
Here it is deciphered:
```

2. Upload your commented code to Canvas as “Lab5_part1.py”.

   Done!

Part 2

1. XOR the known plaintext with the first X bits of the ciphertext, where X is the number of bits in the known plaintext

110101111000100

2. Calculate the period by showing the number of bits that are repeated in the keystream

   After looking at the suspected keystream, we find that the period is 15. In particular, we found this repeating set: 110101111000100

3. Calculate the degree (Show your work step by step)
   Using what we learned in class:

   Deg = log(base2)(period+1)

   Deg = log(base2)(16)

   Deg = 4.

4. Calculate the seed by pasting your keystream into a spreadsheet and tracing it back to the initial starting values

   Here we find that our key is 1011 as seen below:

	A	B	C	D
	1	0	1	1
2	0	1	0	1
3	1	0	1	0
4	1	1	0	1
5	1	1	1	0
6	1	1	1	1
7	0	1	1	1
8	0	0	1	1
9	0	0	0	1
10	1	0	0	0
11	0	1	0	0
12	0	0	1	0
13		0	0	1
14			0	0
15				0

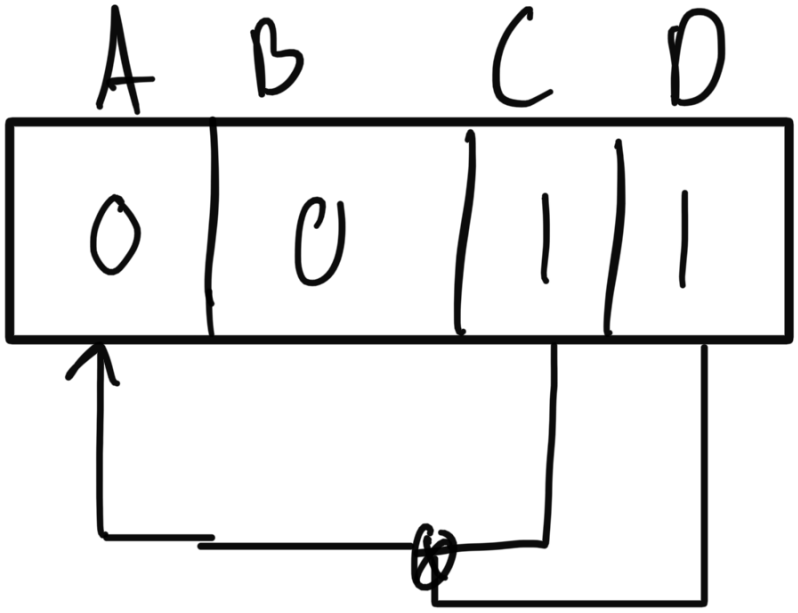
5. Use the spreadsheet from step 4 to find the LFSR configuration and show your work for each step.

   Working our way down the spreadsheet, we come to find rather quickly that the C and D bits influence the next A bit, as seen below:

1	1	0	1	1
2	0	1	0	1
3	1	0	1	0
4	1	1	0	1
5	1	1	1	0
6	1	1	1	1
7	0	1	1	1
8	0	0	1	1
9	0	0	0	1
10	1	0	0	0
11	0	1	0	0
12	0	0	1	0
13		0	0	1
14			0	0
15				0

6. Draw a diagram of the LFSR configuration you identified in step 5

With the previous findings, we come to the following LFSR configuration:



7. Translate the plaintext into ASCII

   I was unable to get the Cycle function down.

8. Upload your code to canvas as “Lab5_part2.py”. A TA will run your program and make sure it can decrypt the ciphertext. Double check that it runs without any issues before uploading it.

   Attached.