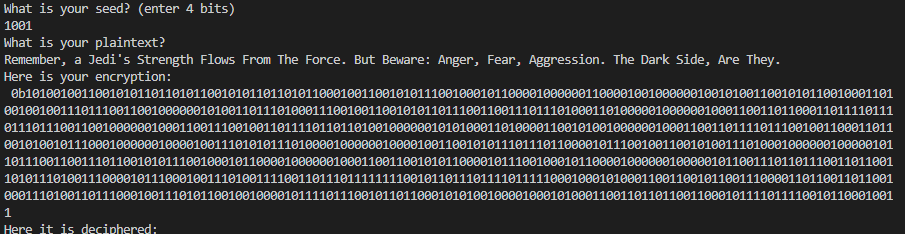
Part 1

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.



1. 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

11010111100010011010111100010011010111100010011010111100010011010111100010011010111100010011010111100010011010

1. 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

1. Calculate the degree (Show your work step by step)

Using what we learned in class:

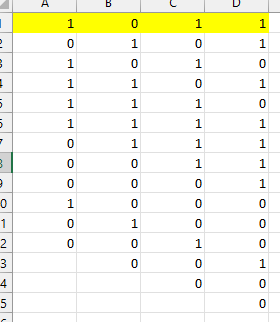
Deg = log(base2)(period+1)

Deg = log(base2)(16)

Deg = 4.

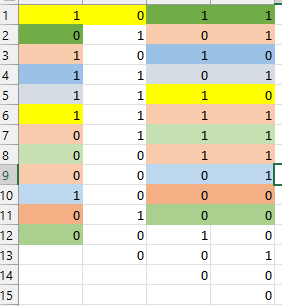
1. Calculate the seed by pasting your keystream into a spreadsheet and tracing it back to the initial starting values

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

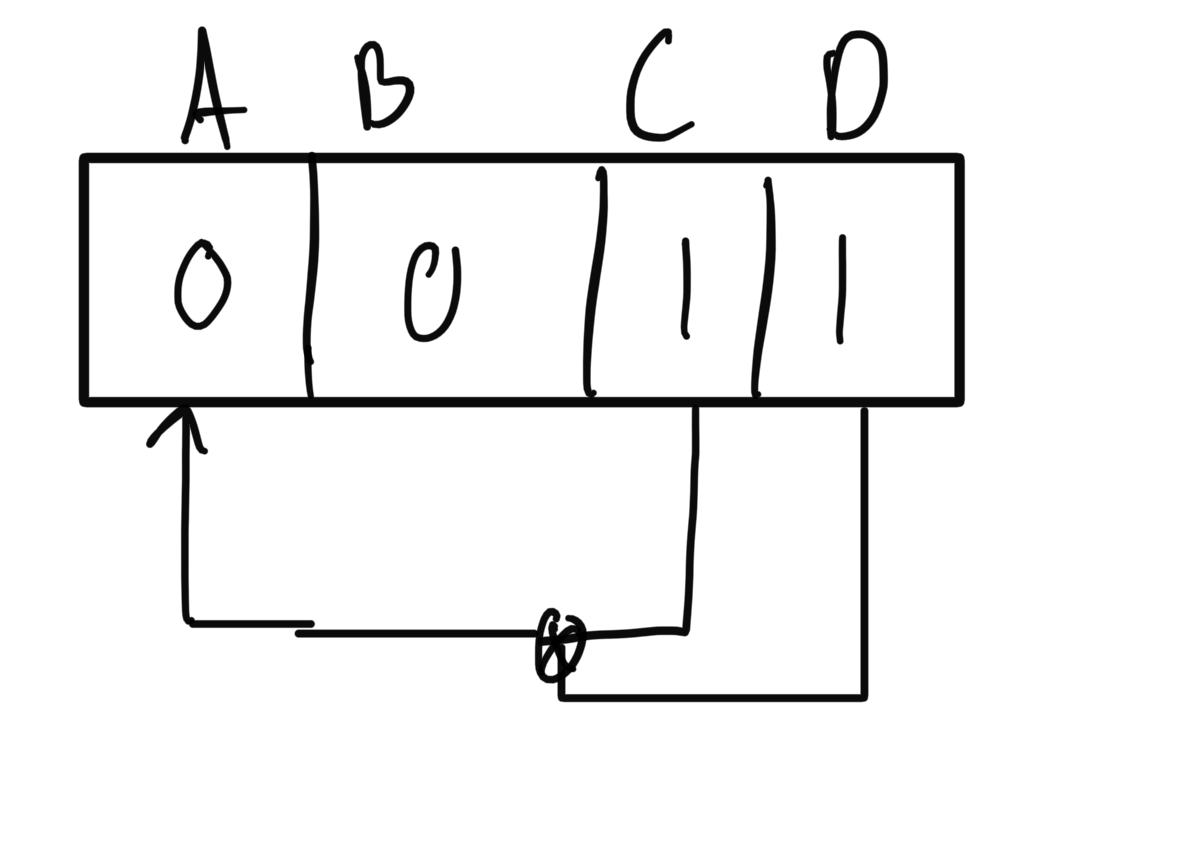


1. Use the spreadsheet from step 4 to find the LFSR configuration and show your work for each step.

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

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

1. Translate the plaintext into ASCII

I was unable to get the Cycle function down.

1. 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.