In this assignment, you need to write a python program to implement Feistel encryption and decryption functions that we started in class. We have limited the scope to only 16 bytes long byte-sequence for this homework, so you won't have to worry about breaking a stream of bytes into 16 bytes each.

Three functions **xor**, **F**, and **gen\_keylist**have been provided to you that are ready to use as they are. There are 3 more functions named **feistel\_block**, **feistel\_enc**, and **feistel\_dec,**with their skeleton already present in the [python notebook](https://umd.instructure.com/courses/1318851/files/66940827?wrap=1)[Download python notebook](https://umd.instructure.com/courses/1318851/files/66940827/download?download_frd=1)we started in class.

Please read this complete description to make sure your program satisfies all the requirements.

### You MUST:

1. Have a global variable named "**UID**" that should have your University ID in **integer**. e.g., UID = 1234

2. Have a global variable named "**Last\_Name**" that should have your last name (as in Canvas) as a **string**. e.g., Last\_Name = 'Last'

3. Have a global variable named "**First\_Name**" that should have your first name (as in Canvas) as a **string**. e.g., First\_Name = 'First'

4. Your program should handle shorter inputs like **b'isthis16bytes?'**by adding **b'\x20'** (byte equivalent of space) so that it becomes a 16 bytes long byte-sequence; For example, **b'isthis16bytes?'** is a 14 bytes long byte-sequence so you'll need to add the space **b'\x20'** twice to make it 16bytes.

5. Have a function with the name "**feistel\_enc**" that accepts **up to** **16 bytes long byte-sequence as its first argument**, **an integer (specifying the number of rounds) as its second argument, and a positive integer (specifying the seed value for the key generation) as its third argument** and **returns a 16 bytes long encrypted byte-sequence**. e.g., function call: **feistel\_enc(b'isthis16bytes?',16,50)** should return a byte-sequence **b'}\xd9\x93-G\x8e\xaa5\x95\x84\n\xb7q\xc4>\xb6'**

6. Have a function with the name "**feistel\_dec**" that accepts **up to** **16 bytes long byte-sequence as its first argument**, **an integer (specifying the number of rounds) as its second argument, and a positive integer (specifying the seed value for the key generation) as its third argument** and **returns a 16 bytes long decrypted byte-sequence**. e.g., function call: **feistel\_dec(b'}\xd9\x93-G\x8e\xaa5\x95\x84\n\xb7q\xc4>\xb6',16,50)** should return a byte-sequence **b'isthis16bytes?  '**or **b'isthis16bytes?\x20\x20'**

7. Shown above is just an example for input string = b'isthis16bytes?', rounds = 16, and seed = 50. Your program should be able to handle all the other variations in values.

8. Write your own test code (if any) in the "\_\_main\_\_" if block and not outside the functions.

9. Submit only a .py file and NOT a .ipynb file.

### You MAY:

1. Write your own extra functions for testing without affecting the functions required for this submission.

2. Rewrite existing supporting functions as long as the expected result is achieved.

3. Use print statements to print out values for your own reference.

4. Use any development environment you're comfortable with as long as you can meet the resulting program's requirements.

5. Use [this sample notebook](https://umd.instructure.com/courses/1318851/files/66940827?wrap=1)[Download this sample notebook](https://umd.instructure.com/courses/1318851/files/66940827/download?download_frd=1)(also accessible through the folder "Homework Format Descriptors" in the Files section on Canvas) to get started, which you may upload on [Google Colaboratory (Links to an external site.)](https://colab.research.google.com/).

6. Name your python (.py) file whatever you want.