In this assignment, you test the avalanche property of AES, i.e., how many bits of ciphertext change if we change just one bit of either the plaintext or the key. 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. You can look at the image\_encryption.pynb file in course files to see examples of how pad and unpad functions can be used if padding is needed. That file also shows you clear examples of encryption and decryption of arbitrary sequences for both DES and AES.

NOTE: The list of bits starts from 0, so 5 in that list would mean the 6th bit and not the fifth bit since we start the count at 0.

You will need to implement a total of two functions**,**with their skeleton already present in [this python notebook](https://umd.instructure.com/courses/1318851/files/66391109?wrap=1)(Part 1 of the notebook).

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. Have a function with the name "**aes\_input\_av\_test**" that accepts **up to16 bytes long byte-sequence (plaintext) as its first argument**, **a 16 bytes long byte-sequence (key) as its second argument,** and a **list (specifying the position of the bit to be flipped) as its third argument**and **returns a list of the same length as the third argument**. e.g., function call: **aes\_input\_av\_test(b'isthis16bytes?',b'veryverylongkey!',[5, 29, 38])** should return a list **[71, 69, 53]**
5. Have a function with the name "**aes\_key\_av\_test**" that accepts **up to16 bytes long byte-sequence (plaintext) as its first argument**, **a 16 bytes long byte-sequence (key) as its second argument,** and a **list (specifying the position of the bit to be flipped) as its third argument**and **returns a list of the same length as the third argument**. e.g., function call: **aes\_key\_av\_test(b'isthis16bytes?',b'veryverylongkey!',[5, 29, 38])** should return a list **[57, 64, 67]**

**Note – For the above two functions, the list represents 3 iterations of bit modifications, ie, for example,  if the 5th bit of the plaintext is flipped, there is a change of 71 bits in the ciphertext and if the 5th bit of the key is flipped there is a change of 57 bits in the ciphertext.**

**For Encryption and Decryption please use the ECB Cipher mode(No Initial Value Required).**

**Hint- Check if padding may be necessary for the plaintext. You can see examples of pad (and unpad) functions in the image encryption notebook that is uploaded on Canvas or just refer to pycryptodome documentation.**

Use the **pycryptodome**library and not any other library for crypto functions.

Write your own test code (if any) in the "\_\_main\_\_" if block AND NOT OUTSIDE THE FUNCTIONS. If code is written outside the functions, marks will be deducted.

**Submit only a single .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. Name your python (.py) file whatever you want.