Hector Lopez Homework 6

1. I used pycrypt, an encryption module for Python https://pypi.org/project/pycrypto/
2.

```
from Crypto.Cipher import AES
import  os

key = os.urandom(16)

iv = os.urandom(16)

aesCBC = AES.new(key, AES.MODE_CBC, iv)
aesECB = AES.new(key, AES.MODE_ECB)

# read file
in_file = open("plaintext.txt", "rb")
# get bytes from file
data = in_file.read()
in_file.close()
# check if total bytes are multiples of 16, add padding as needed
mod = len(data) % 16
if mod:
   data += bytes(16 - mod)

ecb = aesECB.encrypt(data)
cbc = aesCBC.encrypt(data)

print("ECB Encrypt => ",ecb.hex(),"\n")
print("CBC Encrypt => ",cbc.hex(),"\n")

aesCBC = AES.new(key, AES.MODE_CBC, iv)
aesECB = AES.new(key, AES.MODE_ECB)

ecb = aesECB.decrypt(ecb)
cbc = aesCBC.decrypt(cbc)

print("ECB Decrypt => \n" + ecb.decode("utf-8") + "\n")
print("CBC Decrypt => \n" + cbc.decode("utf-8") + "\n")
```

3. ECB Encrypt =>
   e369cb79ea2e54a1ea74a47140e2ef4ee369cb79ea2e54a1ea74a47140e2ef4ee369cb79ea2e54a1ea74a47140e2ef4ee369cb79ea2e54a1ea74a47140e2ef4ee471e5b265dbc2346cfcd06d2b8030ea1f3a5e8a83c6942f5e4487a07b8aeaf0

   CBC Encrypt =>
   ebff67ec33534fdb2336dd3fbb039fff96bfec37b3d09bef870affcc04d58e69e2bfd579ee4c

2de971990fce294ec83c616233facf2252e33568d0fdd7f6e18bc0f25bfb7c5cd51a9ef0c9b
61ff610fdd62395efb955c31bbb2426f995cebc6b

ECB Decrypt =>
GGGGGGGGGGGGGGG
GGGGGGGGGGGGGGG
GGGGGGGGGGGGGGG
GGGGGGGGGGGGGGG
AAAAAAAAAAAAAAA
C

CBC Decrypt =>
GGGGGGGGGGGGGGG
GGGGGGGGGGGGGGG
GGGGGGGGGGGGGGG
GGGGGGGGGGGGGGG
AAAAAAAAAAAAAAA
C

4.
   a. It was much easier to use a library to use aes encryption.
   b. When learning how to use the library, it was easy to understand the purpose of the key and initialization vector.
   c. Implementing AES also helped to understand what was being returned when doing encryption/decryption and be able to modify it so it was legible.