READ THIS FIRST:
Do your best to do every item on your own; if you cannot immediately do an item, go on to others and then come back to it later. Please check the resources section if you have any problems and talk with your professor if there are any further questions.

Due: Wednesday, April 19, 2017.

Goals:

- Practice getting around the command line compiling and running Java programs.
- Practice getting around in and using GitHub.
- Explain some key concepts of the Advanced Encryption System (AES).
- To get you familiar with one of the best cryptosystem of our time.
- Work harder for lab points.

Description:

It is public domain information that the U.S. Government allows usage of the Advanced Encryption Standard (AES) to protect SECRET and TOP SECRET information depending on the key-length used. You have already developed the part of AES that produces keys for every round of encryption. See Lab 4 for details. This laboratory assignment builds on Lab 4 and continues the development of the functions of AES.

You will use your previously created AES files that included: Driver.java and AEScipher.java. In your AEScipher.java file, you will add the following new methods:

1. **Method for AES Add Key.** Write a method with syntax `outStateHex = AESStateXOR(sHex, keyHex)` whose inputs and output are four by four matrices where every element is a pair of hex digits and that will perform the “Add Round Key” operation; that is, the entries of the output matrix are simply the XOR of the corresponding input matrix entries. Here is a test case:

   \[
   \begin{bmatrix}
   54 & 4F & 4E & 20 \\
   77 & 6E & 69 & 54 \\
   6F & 6E & 77 & 20 \\
   20 & 20 & 65 & 6F \\
   \end{bmatrix}
   \oplus
   \begin{bmatrix}
   54 & 73 & 20 & 67 \\
   68 & 20 & 4B & 20 \\
   61 & 6D & 75 & 46 \\
   74 & 79 & 6E & 75 \\
   \end{bmatrix}
   =
   \begin{bmatrix}
   00 & 3C & 6E & 47 \\
   1F & 4E & 22 & 74 \\
   0E & 08 & 1B & 31 \\
   54 & 59 & 0B & 1A \\
   \end{bmatrix}
   \]

2. **Method for AES Nibble Substitution.** Write a method with the following desired syntax: `outStateHex = AESNibbleSub(inStateHex)`. The method’s input and output are 4 by 4 matrices of pairs of hex digits. The method will perform the “Substitution” operation, i.e., the entries of the output matrix
result from running the corresponding input matrix entries through the AES S-Box. Hint: you should use the method you created in Lab 4 \texttt{outHex = AESSBox\texttt{\((inHex)\)}} somehow in this method. Here is a test case:

\[
\begin{bmatrix}
00 & 3C & 6E & 47 \\
1F & 4E & 22 & 74 \\
0E & 08 & 1B & 31 \\
54 & 59 & 0B & 1A
\end{bmatrix}
\text{ produces the following output }
\begin{bmatrix}
63 & EB & 9F & A0 \\
C0 & 2F & 93 & 92 \\
AB & 30 & AF & C7 \\
20 & CB & 2B & A2
\end{bmatrix}
\] (2)

3. Method for AES Shift Rows. Write a method with syntax \texttt{outStateHex = AESShiftRow\texttt{\((inStateHex)\)}} whose inputs and output are 4 by 4 matrices of pairs of hex digits and will perform the Shift Row operation of the AES to transform the input state matrix into output state. Here is a test case:

\[
\begin{bmatrix}
63 & EB & 9F & A0 \\
C0 & 2F & 93 & 92 \\
AB & 30 & AF & C7 \\
20 & CB & 2B & A2
\end{bmatrix}
\text{ produces the following output }
\begin{bmatrix}
63 & EB & 9F & A0 \\
2F & 93 & 92 & C0 \\
AF & C7 & AB & 30 \\
A2 & 20 & CB & 2B
\end{bmatrix}
\] (3)

4. Method for AES Mix Column. Write a method with the following desired syntax: \texttt{outStateHex = AESMixColumn\texttt{\((inStateHex)\)}}. The method’s input and output are 4 by 4 matrices of pairs of hex digits and will perform the Mix Column operation of AES to transform the input state into output state. Here is a test case:

\[
\begin{bmatrix}
63 & EB & 9F & A0 \\
2F & 93 & 92 & C0 \\
AF & C7 & AB & 30 \\
A2 & 20 & CB & 2B
\end{bmatrix}
\text{ produces the following output }
\begin{bmatrix}
BA & 84 & E8 & 1B \\
75 & A4 & 8D & 40 \\
F4 & 8D & 06 & 7D \\
7A & 32 & 0E & 5D
\end{bmatrix}
\] (4)

This one is a little tricky because it is doing multiplications in the Galois fields. So, this item requires from you to do some graduate level research. But do not worry, I have provided great resources below in the -resources- section.

5. Method for AES Encryption. Write a method with syntax \texttt{cTextHex = AES\texttt{\((pTextHex, keyHex)\)}} that will perform AES encryption following the algorithm we discussed in class and shown in Figure 1. Here is a test case; for the following key:

\[
5468617473206D79204B756E67204675
\]

and the following plaintext:

\[
54776F204F6E5204E696E652054776F
\]

the output should be:

\[
29C3505F571420F6402299B31A02D73A
\]

Intuitively, this method will make use of all the methods you have previously developed, so, make sure everything is properly tested.

Your mission is to write the Java programs above, and also you must submit at least three test cases, named \texttt{test.1.txt}, \texttt{test.2.txt}, and \texttt{test.3.txt}. These test cases must be different from the ones found
commonly on-line. You can (and should) use test vectors from the web, but you will then create your own and submit those.

Your Driver.java program will test your implementation by calling AES() providing valid data.

---

Input:

Your driver should read the system key, $K_e$, and plaintext block $p$ from standard input, i.e., System.in. The key and plaintext block should be all in upper case.

Output:

The output must be the ciphertext, all in upper case.

Sample Input 1:

```
5468617473206 D79204B756E67204675
54776F204F6E65204E696E652054776F
```
Sample Output 1:

```
29C3505F571420F6402299B31A02D73A
```

**Resources:**

- Your textbook (Stanoyevitch)!
- Project submission guidelines for this course: [www.reev.us/mscs630s17/project_submission.html](http://www.reev.us/mscs630s17/project_submission.html)
- Coding style guidelines for this course: [www.reev.us/mscs630s17/style.html](http://www.reev.us/mscs630s17/style.html)
- “How to” use the command line “shell”: [www.reev.us/mscs630s17/shell.html](http://www.reev.us/mscs630s17/shell.html)
- Stack Overflow Java Tag: [http://stackoverflow.com/questions/tagged/java](http://stackoverflow.com/questions/tagged/java)

**Submission:**

Submit the following files

1. Driver.java
2. AEScipher.java
3. test.1.txt
4. test.2.txt
5. test.3.txt

- Push your work to your GitHub repository before the due date (see the top of this document). Remember to include your name, the date, and the assignment in the (copious, meaningful, and accurate) commit messages. Then **double check your files are on GitHub**.
- Make sure that your program (*.java) is in a folder called `labs/5/` inside your repository folder.