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, also ask your professor and post your question in piazza.

Due: The Sunday following the Sunday after posted.

Goals:

- Practice getting around the command line compiling and running Java programs.
- Practice getting around in and using the lab submission site.
- Explain some key concepts of the Advanced Encryption System (AES).
- To get you familiar with one of the best cryptosystems of our time.
- Work hard 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. One important part of the AES is how it produces keys on every round of encryption. The instructions on how to generate the 11 round keys K_i, for $i = 0, 1, \ldots, 10$, from the original 128-bit key K_e are the following:

1. Let $K_e \in \mathbb{N}_{\text{hex}}^{4\times4}$ be the 128-bit encryption key in the form of a 4×4 matrix as follows:

$$K_e = \begin{bmatrix} k_{0,0} & k_{0,1} & k_{0,2} & k_{0,3} \\ k_{1,0} & k_{1,1} & k_{1,2} & k_{1,3} \\ k_{2,0} & k_{2,1} & k_{2,2} & k_{2,3} \\ k_{3,0} & k_{3,1} & k_{3,2} & k_{3,3} \end{bmatrix} \tag{1}$$

where every entry $k_{i,j} \in \mathbb{N}_{\text{hex}}$ contains a pair of hexadecimal digits, e.g. $k_{1,2} = \texttt{E0}$. Similarly, let $W \in \mathbb{N}_{\text{hex}}^{4\times44}$ be a 4×44 matrix of the following form:

$$W = \begin{bmatrix} w_{0,0} & w_{0,1} & w_{0,2} & \cdots & w_{0,43} \\ w_{1,0} & w_{1,1} & w_{1,2} & \cdots & w_{1,43} \\ w_{2,0} & w_{2,1} & w_{2,2} & \cdots & w_{2,43} \\ w_{3,0} & w_{3,1} & w_{3,2} & \cdots & w_{3,43} \end{bmatrix} \tag{2}$$

where every entry $w_{i,j} \in \mathbb{N}_{\text{hex}}$ contains a pair of hexadecimal digits, e.g. $w_{1,2} = \texttt{9A}$. Then, let $k(i), w(j) \in \mathbb{N}_{\text{hex}}^4$ denote column vectors constructed from the i-th column of K_e and j-th column of W, respectively as follows:

$$k(i) = \begin{bmatrix} k_{0,i} \\ k_{1,i} \\ k_{2,i} \\ k_{3,i} \end{bmatrix}, \quad w(j) = \begin{bmatrix} w_{0,j} \\ w_{1,j} \\ w_{2,j} \\ w_{3,j} \end{bmatrix}. \tag{3}$$

Table 1: S-box to transform bytes. For example, if the input byte is `8B` the corresponding transformed output byte will be `3D`, because you take 8 as the row and B as the column in the table, which points to `3D`.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

2. Now, we take the AES key and make it be the first four columns of W by making $w(0) = k(0)$, $w(1) = k(1)$, $w(2) = k(2)$, and $w(3) = k(3)$. That creates the first four columns of W, and that constitutes round $i = 0$.

3. For the other 40 columns we do the following:

   (a) If the colum index j is not a multiple of 4. We `XOR` the fourth past and last column with respect to j, as denoted in the following equation:

   $$w(j) = w(j-4) \oplus w(j-1) \tag{4}$$

   (b) If the colum index j is a multiple of 4, this indicates that we are starting a new round i, but we can always know in what round we are by computing $i = \lfloor \frac{j}{4} \rfloor$; and we proceed as follows:

      i. For the construction of $w(j)$ we will use the elements of the previous column $w(j-1) = \begin{bmatrix} w_{0,j-1} & w_{1,j-1} & w_{2,j-1} & w_{3,j-1} \end{bmatrix}^T = w_{\text{new}}$ and store them into a temporary vector w_{new}.

      ii. Then we perform a shift to the left as follows: $w_{\text{new}} = \begin{bmatrix} w_{1,j-1} & w_{2,j-1} & w_{3,j-1} & w_{0,j-1} \end{bmatrix}^T$.

      iii. Next we transform each of the four bytes in w_{new} using an S-box function $S(\cdot)$ (supported by Table 1) as follows $w_{\text{new}} = \begin{bmatrix} S(w_{1,j-1}) & S(w_{2,j-1}) & S(w_{3,j-1}) & S(w_{0,j-1}) \end{bmatrix}^T$.

      iv. Get the Rcon(i) constant for the i-th round by using the look-up Table 2.

      v. Perform an `XOR` operation using the corresponding round constant obtained in the previous step as follows: $w_{\text{new}} = \begin{bmatrix} (\text{Rcon}(i) \oplus S(w_{1,j-1})) & S(w_{2,j-1}) & S(w_{3,j-1}) & S(w_{0,j-1}) \end{bmatrix}^T$.

      vi. Finally, $w(j)$ can be defined as follows:

      $$w(j) = w(j-4) \oplus w_{\text{new}}. \tag{5}$$

4. Every round key is then composed of 4 successive readings of the columns of W. E.g., round zero is composed of $w(0)$, $w(1)$, $w(2)$, and $w(3)$; round one will be composed of $w(4)$, $w(5)$, $w(6)$, and $w(7)$; and so on.

Table 2: Data to retrieve the i-th round constant Rcon(i). For example, Rcon(1) = 1, the Rcon(2) = 2, Rcon(3) = 4, and Rcon(9) = 1B.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
8D	01	02	04	08	10	20	40	80	1B	36	6C	D8	AB	4D	9A
2F	5E	BC	63	C6	97	35	6A	D4	B3	7D	FA	EF	C5	91	39
72	E4	D3	BD	61	C2	9F	25	4A	94	33	66	CC	83	1D	3A
74	E8	CB	8D	01	02	04	08	10	20	40	80	1B	36	6C	D8
AB	4D	9A	2F	5E	BC	63	C6	97	35	6A	D4	B3	7D	FA	EF
C5	91	39	72	E4	D3	BD	61	C2	9F	25	4A	94	33	66	CC
83	1D	3A	74	E8	CB	8D	01	02	04	08	10	20	40	80	1B
36	6C	D8	AB	4D	9A	2F	5E	BC	63	C6	97	35	6A	D4	B3
7D	FA	EF	C5	91	39	72	E4	D3	BD	61	C2	9F	25	4A	94
33	66	CC	83	1D	3A	74	E8	CB	8D	01	02	04	08	10	20
40	80	1B	36	6C	D8	AB	4D	9A	2F	5E	BC	63	C6	97	35
6A	D4	B3	7D	FA	EF	C5	91	39	72	E4	D3	BD	61	C2	9F
25	4A	94	33	66	CC	83	1D	3A	74	E8	CB	8D	01	02	04
08	10	20	40	80	1B	36	6C	D8	AB	4D	9A	2F	5E	BC	63
C6	97	35	6A	D4	B3	7D	FA	EF	C5	91	39	72	E4	D3	BD
61	C2	9F	25	4A	94	33	66	CC	83	1D	3A	74	E8	CB	8D

Your mission is to write a Java program in two files, `Driver_lab4.java` and `AESCipher.java`. The first file, `AESCipher.java`, will have a class for the AES cipher that in this case implements a method with the following signature `String[] roundKeysHex = aesRoundKeys(String KeyHex)` that will produce 11 round keys (one in every element of the string array) as explained in the above **Description** section. The input, `KeyHex`, is a length 16-hex string representation of the system key K_e. The output, `roundKeysHex`, will be an 11-row string array representation of all the round keys. Each element of `roundKeysHex` will contain a 16-hex string corresponding to each round key. You will also create two more methods to help you in yor computations, one for reading the S-box with a signature `outHex = aesSBox(inHex)`, and another method to get each round's constant with the following signature `outHex = aesRcon(round)`. So, your `AESCipher.java` file will have the implementation of all your AES code.

However, your `Driver_lab4.java` program will only test your implementation by calling `aesRoundKeys()` providing valid data.

The program should produce only keys as shown in the example below. Above all things try to make your code as efficient as possible. Look at the resources section below to copy-paste tables. Whenever possible try to work directly with hexadecimal values in Java.

Input:

Your driver should read the system key, K_e, from standard input, i.e., `System.in`. The key should be all in upper case

Output:

The output must be the 11 round keys, one in each row, all in upper case.

Sample Input 1:

```
5468617473206D79204B756E67204675
```

Sample Output 1:

```
5468617473206D79204B756E67204675
E232FCF191129188B159E4E6D679A293
56082007C71AB18F76435569A03AF7FA
D2600DE7157ABC686339E901C3031EFB
A11202C9B468BEA1D75157A01452495B
B1293B3305418592D210D232C6429B69
BD3DC287B87C47156A6C9527AC2E0E4E
CC96ED1674EAAA031E863F24B2A8316A
8E51EF21FABB4522E43D7A0656954B6C
BFE2BF904559FAB2A16480B4F7F1CBD8
28FDDEF86DA4244ACCC0A4FE3B316F26
```

Resources:

- Your textbook (Stanoyevitch)!
- Project submission guidelines for this course (posted on iLearn)
- Coding style guidelines for this course (posted on iLearn)
- "How to" use the command line "shell" (posted on iLearn)
- Piazza for asking questions to professor and classmates use the tag: `lab4`
- The official Java reference: `http://docs.oracle.com/javase/tutorial/collections/TOC.html`
- Stack Overflow Java Tag: `http://stackoverflow.com/questions/tagged/java`
- General info about AES `https://en.wikipedia.org/wiki/Advanced_Encryption_Standard`
- General info about key scheduling: `https://en.wikipedia.org/wiki/Rijndael_key_schedule`
- General info about AES S-box: `https://en.wikipedia.org/wiki/Rijndael_S-box`

Submission:

- Upload your work to the submission site `https://car.rivas.ai` and submit your `Driver_lab4.java` and `AESCipher.java` before the due date and make sure they pass all the tests. If they do not pass all the tests, then it means that your programs are incorrect and you need to keep working on them.
- Once you pass all the tests, your professor will review your code for style and then you will receive a grade.