DES in Java

In this assignment you will use the DES algorithm offered by Java to encrypt and decrypt data. Furthermore, I'll give you a small code-breaking challenge.

Implementation Aspects

The Java SDK has a set of cryptographic tools which can be used in any Java application. For more information, see the Java Cryptography Architecture (JCA) Reference Guide at
http://docs.oracle.com/javase/7/docs/technotes/guides/security/crypto/CryptoSpec.html#Cipher

However, this reference contains a lot more detail than you will need for this project; the sample file (DESExample.java) should be sufficient. Note that in a DES key, the low bit of each byte is ignored.

Team Work and Honor Code

You will work in teams of two students and you are allowed to discuss the project in general terms with other teams.
HOWEVER, you are absolutely not allowed to share code or results with other teams. By "sharing code" I do not only mean actually copying/pasting the code, but also writing/printing it on paper and showing it to someone else, or simply showing someone the code on screen. Violation of these rules will be treated as an honor code violation!

Deliverables

Submit your answers to the questions below as well as your code through CNU Scholar. Make sure that your code (and any other document you submit) contains a comment with the names of all team members. If you like, you can include your answers as comments in the .java file; if you do this, put all answers together in a single block of comments at the beginning of the file (i.e. I will not hunt through your code looking for comments that might be construed as answers to the questions!).

Questions

1. Encryption

Given are:

- 8 characters plaintext in ASCII (includes a space but not quote marks): “Dee Bugg”

  Note: This is exactly one block size. Therefore, you only need to call the DES encryption method once.

- Key bytes (also specified as hex numbers): 7A, 90, C8, 36, 44, 0E, 18, 76

  In order to enter these byte values in Java, you'll need to cast each value to (byte) and you have to put a "0x" in front of each number (as in DESExample.java).

What is the ciphertext specified in hex format?

2. Decryption

Given are:

- 8 characters of ciphertext: 9D, 1C, 1D, 94, 8F, 21, 55, C5

  Note: This is again exactly one block size.
- Key bytes: 46, AA, 20, 1E, F4, 3C, 92, D2

What is the decoded plaintext? Specify it as a character string (using ASCII).
Note: If everything worked, the plaintext should be really plain text, i.e. readable.

3. Finding the Key

Given are:

- Plaintext in ASCII: Captains
- The corresponding ciphertext: A5, 99, 04, 72, 39, 95, 41, EC
- The first four bytes of the Key: 90, 4E, F2, CC

What is the complete key used in this encryption?

4. Code Breaking

Here is the story:

- You have overheard following encrypted message given in hex format: B1, 80, E8, 05, 4E, 7D, D6, 4C
- You know that the message is readable text containing only letters (upper and lower case) and spaces.
- Finally, your team of spies has managed to patch together some shards from a paper shredder to recover the first five bytes of the key. They are: BA, 54, 68, 08, 12.

What is the complete message in text format (ASCII)? What is the rest of the key?

Here is a function you might find useful:

```
public static void incrementByteArrayByOne(byte [] b) {
// starting from back, add one and check if carry over
 // if no carry over, stop and return
 // if carry over, increment next byte and check again
        for(int i = b.length-1; i>=0  ; i--){
                b[i]++;
                if(b[i] != 0x00){
                return;
                }
        }
        return;
}
```