Data Preparation

Background

The Feistel method you developed [previously] receives a 6-bit value to be processed. But, characters in our system will be stored as 8-bit values. Thus, to process a string of characters you will need to divide/combine adjacent [in the string] characters to form 6-bit quantities. If we select 3 characters at a time, we will have 24 bits which will result in 4 calls to the Feistel method, 6 bits each. A similar method is required to divide the resultant 12-bit values back into 8-bit values.

Specification

Write a method called

```
public static short[] preprocess(String s)
```

to perform the following operations:

1. Convert the **String** s into an array of **byte**
2. Ensure the length of the array of **byte** is a multiple of 3
   a. If it is not, add values of 0 to the end of the array
3. Combine 3 consecutive **byte** values into two 12-bit values using bit manipulation expressions
   e.g.
   $X_7X_6X_5X_4X_3X_2X_1X_0$ $Y_7Y_6Y_5Y_4Y_3Y_2Y_1Y_0$ $Z_7Z_6Z_5Z_4Z_3Z_2Z_1Z_0$ ->
   $X_7X_6X_5X_4X_3X_2X_1X_0Y_7Y_6Y_5Y_4$ $Y_3Y_2Y_1Y_0Z_7Z_6Z_5Z_4Z_3Z_2Z_1Z_0$
4. Store the 12-bit values into an array of **short** (in the least significant 12-bits)
5. Do this for the entire **byte** array
6. Return the array of **short**

Write a method called

```
public static byte[] postprocess(short[] s)
```

to perform the following operations:

1. Split 2 consecutive **short** values into three 8-bit values using bit manipulation expressions
   e.g.
   $X_7X_6X_5X_4X_3X_2X_1X_0Y_7Y_6Y_5Y_4$ $Y_3Y_2Y_1Y_0Z_7Z_6Z_5Z_4Z_3Z_2Z_1Z_0$ ->
   $X_7X_6X_5X_4X_3X_2X_1X_0$ $Y_7Y_6Y_5Y_4Y_3Y_2Y_1Y_0$ $Z_7Z_6Z_5Z_4Z_3Z_2Z_1Z_0$
2. Store the 8-bit values into an array of **byte**

3. Do this for the entire **short** array
4. Return the array of **byte**

Use the following main method to demonstrate your code:

```java
public static void main (String[] args) {
    short[] data12;
    byte[] data8;

    data12 = preprocess("A");
    for (int i = 0; i < data12.length; ++i) {
        System.out.print(int2binary(data12[i], 12) + " ");
    }
    System.out.println();
    data8 = postprocess(data12);
    for (int i = 0; i < data8.length; ++i) {
        System.out.print(int2binary(data8[i], 8) + " ");
    }
    System.out.println();
    System.out.println();

    data12 = preprocess("AB");
    for (int i = 0; i < data12.length; ++i) {
        System.out.print(int2binary(data12[i], 12) + " ");
    }
    System.out.println();
    data8 = postprocess(data12);
    for (int i = 0; i < data8.length; ++i) {
        System.out.print(int2binary(data8[i], 8) + " ");
    }
    System.out.println();
    System.out.println();

    data12 = preprocess("ABC");
    for (int i = 0; i < data12.length; ++i) {
        System.out.print(int2binary(data12[i], 12) + " ");
    }
    System.out.println();
    data8 = postprocess(data12);
    for (int i = 0; i < data8.length; ++i) {
        System.out.print(int2binary(data8[i], 8) + " ");
    }
    System.out.println();
    System.out.println();

    data12 = preprocess("ABCD");
    for (int i = 0; i < data12.length; ++i) {
        System.out.print(int2binary(data12[i], 12) + " ");
    }
    System.out.println();
    data8 = postprocess(data12);
    for (int i = 0; i < data8.length; ++i) {
        System.out.print(int2binary(data8[i], 8) + " ");
    }
}
```

```java
        System.out.println();
        System.out.println();

        data12 = preprocess("ABCDEF");
        for (int i = 0; i < data12.length; ++i) {
            System.out.print(int2binary(data12[i], 12) + " ");
        }
        System.out.println();
        data8 = postprocess(data12);
        for (int i = 0; i < data8.length; ++i) {
            System.out.print(int2binary(data8[i], 8) + " ");
        }
        System.out.println();
        System.out.println();

    }
```

Deliverables

- Source code (.java) files
- Screen shot of your running program using the main function given (above)
- Reflective essay describing
    - Successes
    - Difficulties (if any) and how you addressed them
    - Lessons learned

If you do this assignment as a programming pair include both names and both reflections in a single document. Each person should submit the requested [identical] documents to Canvas under their own name.