BIT STUFFING EXERCISE J DLC1

PURPOSE

This purpose of this exercise is for you to demonstrate your understanding of bit stuffing by writing methods of stuff and unstuff a bit set.

EXERCISE PROCESS

Begin by installing JNWS as described in the User’s Guide. After confirming it will compile and run, create a new Java class BitStuffing within package Exercises. (To do this in NetBeans, right-click on the Project containing JNWS and select new->class.)

You are provided with a template for your program in class Exercises.BitStuffing. The template also contains the algorithm for bit stuffing, from Understanding Internet Protocols. You can paste this into the workspace in your IDE. The class has four methods

/**Reads a single line from standard input*/
    private static String ReadSingleLineFromStandardIO ( )

/**Stuffs a string of bits*/
    private static Utility.JNWBitSet StuffBitSet ( Utility.JNWBitSet unstuffedBits )

/**Unstuffs a string of bits*/
    private static Utility.JNWBitSet UnstuffBitSet ( Utility.JNWBitSet stuffedBits )

/**Runs a demonstration of bit packing.*/
    public static void main ( String [ ] args )

Complete the method StuffBitSet which accepts an unstuffed bit set and returns the stuffed bit set and the method UnstuffBitSet which accepts a stuffed bit set and returns its unstuffed form.

Suggestions:

1. If you have a bit set of length n bits, what is the maximum possible size of the stuffed bit set? Start by making a stuffed bit set of that size.
2. Since even simple Java IO can become very complicated, you should use the provided method ReadSingleLineFromStandardIO() to read a String representing the email from the standard input.

4. The methods of Utility.JNWBitSet will convert and display bitsets, which are Java vectors that hold the index numbers of positions with true (1) bits. (The lowest index, 0, is at the right end of the BitSet, although that might not be evident when you use System.out.println() to display it.) Methods you are likely to need in that class are:
   - setJNWLength(int length) sets the maximum number of bits in the BitSet
   - getJNWLength() returns the maximum number of bits in the BitSet
   - set(int bitIndex, value) sets index position to the boolean value
   - getBitValue(int bitIndex) returns int 0 or 1 value of the bit at index position
   - shift(int numberOfBitsToShift) shifts the BitSet right (positive) or left (negative)
   - setJNWBitSetToString(String chars) makes characters into a BitSet
   - convertJMBitSetToString() makes a character string from a BitSet
   - writeBitSet() makes a string of 1 and 0 for printing, from a BitSet

5. A good test is to use the input String “00011111111000” in the method main(), without the email conversion.

6. Note that Utility.JNWBitSet has a constructor that accepts a String of 0s and 1s. So you should be able to create the bit set very easily as a variation on your program for the email message.

7. If you use NetBeans, note the advice in the User’s Guide about setting the main() method to be run.

DEMONSTRATION OF WORKING CODE

The class file for Demonstrations.BitStuffing is available for you to see a demonstration. Put the file BitStuffing.jar in a directory and use the command prompt

```
java –jar BitStuffing.jar
```

to run the demonstration, which accepts a sequence of 0 and 1 characters and displays the stuffed version of the sequence.

TESTING YOUR CODE
Write a main() program that accepts as input a String “email”, displays it, converts it to a BitSet, stuffs it, displays the result, unstuffs it, converts back to text, and displays that. Run the program for with email input string “Chicken Little was right~”. (This input will be used by the grader to test your code.)

SUBMITTING

Send your Exercises.BitStuffing.java as an email attachment to the grader.