**Assignment 6**

**Overview**

For this assignment, you will develop Java programs and methods to solve several problems. For some problems, you will also write test cases with which to test your code.

We have created "skeleton" Java files which you will fill in. These are linked in the "Attachments" section, below. Download these files and fill in code where specified. **You should test your methods by writing code in main() (either in the same file or a different file) to call your methods and execute your chosen test cases.**

If you know some Java already, you may use advanced features that we have not covered in class yet (though it is not necessary to do so). However, you may NOT use any modules or classes from outside java.lang (do not add any "import" statements to your programs).

**Input, output, and error conditions**

When writing a *method* , inputs are provided directly through the method's arguments. Output should be returned using the return statement. Error conditions should be reported by returning an appropriate value (as described in the problem). Do not use the IO module.  
Links to Documentation

[Character](http://java.sun.com/javase/6/docs/api/java/lang/Character.html) module:

* [isDigit](http://java.sun.com/javase/6/docs/api/java/lang/Character.html#isDigit%28char%29)
* [isLetter](http://java.sun.com/javase/6/docs/api/java/lang/Character.html#isLetter%28char%29)
* [toLowerCase](http://java.sun.com/javase/6/docs/api/java/lang/Character.html#toLowerCase%28char%29)
* [toUpperCase](http://java.sun.com/javase/6/docs/api/java/lang/Character.html#toUpperCase%28char%29)

[String](http://java.sun.com/javase/6/docs/api/java/lang/String.html) class:

* [charAt](http://java.sun.com/javase/6/docs/api/java/lang/String.html#charAt%28int%29)
* [compareTo](http://java.sun.com/javase/6/docs/api/java/lang/String.html#compareTo%28java.lang.String%29) (alphabetical order, case-sensitive)
* [compareToIgnoreCase](http://java.sun.com/javase/6/docs/api/java/lang/String.html#compareToIgnoreCase%28java.lang.String%29) (alphabetical order, case-insensitive)
* [concat](http://java.sun.com/javase/6/docs/api/java/lang/String.html#concat%28java.lang.String%29) (+)
* [endsWith](http://java.sun.com/javase/6/docs/api/java/lang/String.html#endsWith%28java.lang.String%29)
* [equals](http://java.sun.com/javase/6/docs/api/java/lang/String.html#equals%28java.lang.Object%29) (case-sensitive)
* [equalsIgnoreCase](http://java.sun.com/javase/6/docs/api/java/lang/String.html#equalsIgnoreCase%28java.lang.String%29) (case-insensitive)
* [indexOf](http://java.sun.com/javase/6/docs/api/java/lang/String.html#indexOf%28int%29) (finds a single character)
* [indexOf](http://java.sun.com/javase/6/docs/api/java/lang/String.html#indexOf%28java.lang.String%29) (finds a string)
* [lastIndexOf](http://java.sun.com/javase/6/docs/api/java/lang/String.html#lastIndexOf%28int%29) (finds a single character)
* [lastIndexOf](http://java.sun.com/javase/6/docs/api/java/lang/String.html#lastIndexOf%28java.lang.String%29) (finds a string)
* [length](http://java.sun.com/javase/6/docs/api/java/lang/String.html#length%28%29)
* [startsWith](http://java.sun.com/javase/6/docs/api/java/lang/String.html#startsWith%28java.lang.String%29)
* [substring](http://java.sun.com/javase/6/docs/api/java/lang/String.html#substring%28int,%20int%29) (substring from start, up to but not including end)
* [substring](http://java.sun.com/javase/6/docs/api/java/lang/String.html#substring%28int%29) (substring to end of string)
* [toLowerCase](http://java.sun.com/javase/6/docs/api/java/lang/String.html#toLowerCase%28%29)
* [toUpperCase](http://java.sun.com/javase/6/docs/api/java/lang/String.html#toUpperCase%28%29)
* [trim](http://java.sun.com/javase/6/docs/api/java/lang/String.html#trim%28%29) (trims spaces from ends of string)

**Problems**

**Problem 1**

Write your code in the file StringRec.java. For this problem, the following restrictions apply:

* YOUR CODE MUST BE RECURSIVE.
* Do not use loops (while, do/while, or for).
* Do not declare any variables outside of a method. You may declare local variables inside a method.

Complete the following method:

* public static String decompress(String compressedText): Decompress the input text, which has been compressed using the RLE algorithm (previous hw assignment):

Run-length encoding (RLE) is a simple "compression algorithm" (an algorithm which takes a block of data and reduces its size, producing a block that contains the same information in less space). It works by replacing repetitive sequences of identical data items with short "tokens" that represent entire sequences. Applying RLE to a string involves finding sequences in the string where the same character repeats. Each such sequence should be replaced by a "token" consisting of:

* 1. the number of characters in the sequence
  2. the repeating character

If a character does not repeat, it should be left alone.

For example, consider the following string:

qwwwwwwwwweeeeerrtyyyyyqqqqwEErTTT

After applying the RLE algorithm, this string is converted into:

q9w5e2rt5y4qw2Er3T

In the compressed string, "9w" represents a sequence of 9 consecutive lowercase "w" characters. "5e" represents 5 consecutive lowercase "e" characters, etc.

You may assume that the character counts will be single-digit numbers (a character will not repeat more than 9 times consecutively).

Hint #1: remember that characters are represented by numeric codes. You can decrement a character variable as follows:

char c = '7';

c--; // c will now hold the character '6'

Hint #2: You probably will not need to use this hint for this problem. However, a fast way to convert a digit character into the numeric value of the digit is to subtract the character code for the digit zero:

char c = '7'; // this has the character code 55, not 7

int x = c - '0'; // this produces the number 7