CPE 102 Lab 9 - Recursion

Goals:

- To gain experience developing recursive algorithms.
- To gain experience developing structurally recursive algorithms.

PART A - Recursive Algorithms

Write a class `RecursionLab.java` that follows this specification.
The **RecusionLab** class contains three static methods that must be implemented recursively to get credit for the lab. ***Hint***: For the `isPalindrome` method, you will need the `substring` method of the **String** class.

Test your code thoroughly. You are provided with a sample tester (see below). Add to this tester!

PART B - Structural Recursion

You will implement a system of classes to represent a ***TotemPole*** for the fictional Gashunonga Tribe. A ***TotemPole*** is compromised of different animial heads (**Bear**, **Snake**, and **Eagle**). Every ***TotemPole*** has an **Eagle** at the top. **Eagles** can only be at the top of the pole. Each animal head adds a different amount of power to the ***TotemPole***. **Bears** are the most sacred to the Gashunongan people and add 5 power points to the ***TotemPole***. **Snakes** add 3 power points. **Eagles** add 2 power points. Since **Bears** are the most sacred, a Gashunongan Chief's ***TotemPole*** must have at least 3 **Bears** in a row at some place in the ***TotemPole***.

Part B Specification:

Develop an interface called ***TotemPole***. A ***TotemPole*** has the following methods:

```
int power();  // the total power of the pole
int height(); // the number of heads in the pole
boolean chiefPole(int bearCount);  // is this pole worthy
of a chief?
                // bearCount is how many bears encountered in a row
so far
```

Develop the class **Bear** that implements the ***TotemPole*** interface. **Bears** have the following constructor:

```
public Bear(TotemPole rest);
```

Develop the class **Snake** that implements the ***TotemPole*** interface. **Snakes** have the following constructor:

```
public Snake(TotemPole rest);
```

Develop the class **Eagle** that implements the ***TotemPole*** interface. **Eagles** have the following constructor:

```
public Eagle();
```

TESTING

Sample JUnit test files for both parts of the lab are posted on PolyLearn. You must pass these tests to get any credit for the lab. Per usual, these files will hardly test any of the functionality of your lab, but are merely to ensure your method names are correct, etc. Add to these files!

HANDIN

```
Submit these files to PolyLearn:

 RecursionLab.java TotemPole.java Bear.java Snake.java Eagle.java
```