LAB S06 – RECURSION INTRO RECURSION

Make a `Driver` class that creates a `Recursion` object (this means you will need a `Recursion` class, too). Your `Recursion` class should contain the following four methods.

`public int sumOfFirstNNumbers( int n )`

This method should recursively calculate the sum of the first `n` positive integers, beginning with 1. For example, the call `sumOfFirstNNumbers(5)` should return the value 15, which is 1 + 2 + 3 + 4 + 5.

`public int factorial( int num )`

This method should return the value of the factorial of the given number. A call to `factorial(5)` should return 120, which is 5*4*3*2*1.

`public int fibValue( int loc )`

This method should return the number in the `loc` position in the Fibonacci sequence. As a reminder, the Fibonacci sequence is a sequence of numbers in which each value is determined by summing the two previous values, resulting in the sequence: 1, 1, 2, 3, 5, 8, 13, …. A call to `fibValue(5)` would return the value 8, which is the number at position 5 (assuming we start at position 0) of the sequence.

`public int collatz( int num )`

The Collatz conjecture is a conjecture in mathematics summarized as: if you start with any positive integer and use a HOTPO (half or triple plus one) transformation, you will eventually reach 1. In simpler terms, if the number is even, you cut it in half (8 becomes 4); if the number is odd, you triple it and add one (5 becomes 16).

Write this method to find the number of transformations it takes to get to a value of one. A call to `collatz(10)` will return a value of 6, because it takes six transformations to get from 10 to 1 (10 -> 5 -> 16 -> 8 -> 4 -> 2 -> 1).

Criteria

1) Each method must be recursive.
2) Pre and post conditions are not required for these methods

Submission

1) Submit your code for this lab on a Googly Doc through Canvas by Sunday, February 23 at 11:59 pm. Alternatively, you may show your lab to Holm in class before the beginning of February break. ***If you submit through Canvas, you must provide a sample output.***