Activity 11 CSS 341
Active Learning Exercises

Names (if present in class):

Recursive Algorithms.

We have seen examples of recursive algorithms for Pi and for the factorial function. In this exercise, your team will develop pseudo code for two functions, recursive and iterative, to evaluate the following *n-term* approximation to a mathematical function:

$$\boldsymbol{Floge(n,x) = 2 * \left[\frac{x}{1} + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \cdots \frac{x^{2n-1}}{2n-1}\right] \approx log_e\left[\frac{1+x}{1-x}\right]}$$

Assume your main program obtains an integer value of *n* and a valid value of x from the user (-1 <x <1). It then calls a function *Floge(n,x)* to obtain the *n-term* approximation to the log function above.

1. **Generate pseudo code** for the vbs function *iFloge (n,x)* in which you use a loop structure to iterate over the *n* terms requested in the approximation. The function *iFloge(n,x)* returns the value of this approximation.

2. **There is also a recursive algorithm for this function.** Examining the series for *Floge(n,x),* write down *Floge(4,x)* in terms of *Floge(3,x).* Do this by writing the two expressions in full, comparing them, and then writing one in terms of the other.

Write the general relationship for *Floge(n,x)* in terms of *Floge(n-1,x):*

Write a pseudo code function *rFloge(n,x)* that uses a recursive algorithm to calculate and return the value of the *n-term* approximation desired. The function *rFloge(n,x)* returns the value of this approximation. Do not use a looping structure in this situation. *Don't forget the base case!*