CSCI 4230 – Assignment 4

Instructions

For this assignment you must write the following functions in Scheme:

1. Write a function called `letterGrade` that takes a number between 0 and 100 and returns a string containing the corresponding letter grade where 90 and above is an A, 80-89 is a B, and so on. Use a `cond` expression to implement this. If the number is out of range, return the string `"Error: out of range"`.
2. Write a function called `countdown` that takes a positive integer and uses a `do` expression to display a sequence of numbers counting down to 1, each on its own line, then displaying the string `"Blastoff!"`.
3. Write a recursive function called `eval-poly` that takes a list of numbers representing the coefficients of a polynomial and a value for x and evaluates the polynomial for the given value of x. The list of coefficients should start with the term of lowest degree and end with the term of highest degree. If any term of intermediate degree is missing from the polynomial it should have a coefficient of zero. For example, the polynomial $x^3 + 4x^2 + 2$ would be represented by the list `'(2 0 4 1)`. Hint: the polynomial above can be rewritten as $2 + x \cdot (0 + x \cdot (4 + x \cdot 1))$
4. Write a tail-recursive version of the previous problem called `eval-poly-tail`. It should call a helper function called `eval-poly-tail-helper` that uses tail recursion to keep a running sum of the terms evaluated so far. You might want to use the `expt` function to take a number to a power.
5. Write a recursive function called `split` that takes a list and returns a list containing two lists, each of which has roughly half the items in the original list. The easiest way to do this is to alternate items between the two lists, so that `(split '(1 2 3 4 5))` would return `'((1 3 5) (2 4))`. I recommend using two base cases: one for an empty list and the other for a list containing one item.
6. Write a recursive function called `merge` that takes two sorted lists of numbers and merges them together into one sorted list containing all of the number in both lists including duplicates.
7. Write a recursive function called `mergesort` that sorts a list by doing the following:
   (a) Use `split` to split the list into two roughly equal-sized partitions.
   (b) Recursively sort both partitions.
   (c) Use `merge` to merge the sorted partitions together.

   Once again you will need two base cases, one for the empty list and the other for a single-element list.

What to Hand In

Implement all of the functions described above in a source file called `yourlastnameAssign4.scm` with your actual last name. Make sure to put your name, CSCI 4230, and Assignment 4 in the comments (comments in Scheme start with `;;`). Upload the source file to D2L to the dropbox called Assignment 4.