CSC 351L – Comparative Programming Languages Lab
Project 8 – Lisp Evaluation and More Recursion
Due: May 6 at 11:59 PM

Purpose:

To investigate Lisp's ability to execute code that is constructed by a function during execution, as well as gain additional experience with recursion.

Description:

For this project, you will write two different versions of two functions.

The first function is a power function that will calculate x^i, where x can be either a real number or an integer, but i must be an integer. In the first version, you will write this recursively, using the classic technique that was probably used when you first saw recursion. In the second version, you will write a function that will use a loop to construct a list containing the proper number of multiplications and then have Lisp evaluate that.

The second function will receive a list of integers and will return a result that is a two element list consisting of the smallest and largest values in the original list. The first version you will do will have two helper functions that find the largest and smallest values that are used to construct the result. In the second version, you must accomplish this task in one recursive function.

Thoughts:

1) The recursive power function is quite easy, but the second is a little harder. One way to accomplish this is to have a local variable that is used to hold the list as you build it, and then use a `do` loop that keeps adding the proper items to this list in the proper quantity. In other words, to calculate 5^3 you could built the list (* 5 5 5) and then evaluate it.
2) Using helper functions to write the largest/smallest function will make that quite easy, but the non-helper version of this problem can be quite involved. Thinking about the following might help:
   - Remember that one way to accomplish a recursive function is to keep making recursive calls until you get to the empty list and then do the real work on the way back "up."
   - If there is one element in a list, it is both the largest and the smallest.
   - When the recursive function call returns, you now have a two element list with the smallest and largest values that are in the cdr of the list. You now only need to consider how the car of the current list might alter this result.

Deliverables:

When you complete the program, you will prepare two things: a project report and a zip file containing all of your code.

The project report must use the format given in the sample file on D2L. This report can be prepared as a text file, MS Word document, or PDF. The project report should not be included in the zip file.

The zip file should include all lisp files. You can create the zip file in one of two ways – using zip on brahe or on your own computer.

The two items must be uploaded to the Project 8 drop box on D2L. No other form of submission will be accepted.