

CSE 3302 Lab Assignment 2

Due October 2, 2013

Goal:

Understanding of elementary list manipulation in Scheme.

Requirements:

1. Write and test a set of Scheme (Racket) functions to evaluate an arithmetic expression of the form (`evalExp` expression).
 - a. The expression will be a nested S-expression consisting of the following forms:
 1. An integer atom.
 2. The atom `sum` followed by zero or more sub-expressions for which the sum is to be computed.
 3. The atom `product` followed by zero or more s-exps for which the product is to be computed.
 4. The atom `min` followed by one or more s-exps for which the minimum is to be computed.
 5. The atom `max` followed by one or more s-exps for which the maximum is to be computed.
 6. The atom `neg` followed by a single s-exp for which the negation is to be computed.
 - b. The result is to be computed as the result of `evalExp`.
3. You may simply include the provided test cases along with your “definitions” to be run with Racket, but leave them out of your final submission.
4. Email your functions, as an attachment, to `sourabh.bose@mavs.uta.edu` by 12:45 p.m. on October 2, 2013.

Getting Started:

1. A driver function similar to the following is useful for processing the test cases:

```
(define (evalExp l)
  (display l)
  (display " ")
  (evalList l))
```

where `evalList` is a function to parse and call functions for each of the forms described above.

2. You may assume the test cases are formatted correctly.
3. Avoid using `set!` . . .