CS 314 Principles of Programming Languages
Due date: Tuesday, 20, beginning of lecture

Problem 1
Write the following functions on lists in Scheme. The semantics of the functions is described through examples.

1. (define replace
   (lambda (atom1 atom2 l)
     ...))
   ...
   (replace 'c 'd '(a(b)(c)(((e e))))))  -->  (a((b)(d)(((e e))))))

2. (define double-flatten
   (lambda (l)
     ...))
   ...
   (double-flatten '(a((b)(c d)(((e))))))  -->  '(a a b b c c d d e e)

3. (define make-single
    (lambda (l)
      ...))
    ...
    (make-single '(a((b)(c c)(((e e e))))))  -->  (a((b)(c)(((e))))))

Problem 2
Show the steps in a (1) call-by-value reduction interpreter, (2) call-by-name reduction interpreter; and (3) call-by-value closure interpreter for the following TINY programs. Use examples on page 12 and 13 in lecture 21 as a template of how to present these steps. You should assume that the initial environment contains the values for the "build-in" functions + and *.

1. ((lambda (x) (+ x x)) ((lambda(y) y) 5))
2. ((lambda (x) (if x 3 x)) (if #t #f #t))
3. ((lambda (x) (* (x 2) 1)) ((lambda (x) (lambda (y) (+ x y))) (* 3 2)))