EXERCISES ON SCHEME

1. Compute the values of the following expressions.

Expression	Value
(car ‘(6 3 2))	
(car (6 3 2))	
(let ((L ‘(6 3 2))) (car L))	
(cdr ‘((6 3) 2 (7 6) 3))	
(car ‘((6 3) 2 (7 6) 3))	
(cons 6 ‘(3 2 8))	
(cons ‘(5 6) ‘(3 2 8))	

2. Draw box diagrams (i.e. graphical representation) for the following Scheme lists:
   a. (((a) b) c (d e))
   b. (b(a ( )) ((c) d (b)) e)

3. Write a function that returns the nth element of a list.
   examples: (nth 2 ‘(a b (c d))) returns b
   (nth 3 ‘(a b (c d))) returns (c d)

4. Write a function that count the number of occurrences of atoms in a list of atoms.
   example (countatom ‘(a b c d e)) is 5

5. Write a function that counts the number of occurrences of atoms at the “top” level in an arbitrary list.
   example (counttopatom ‘(a (b c) d a)) returns 3

6. Write a Scheme function with two parameters, an atom and a list, that returns the list with all occurrences, no matter how deep, of the given atom deleted. The returned list cannot contain anything in place of the deleted atom. For example:
   (removeatom a ‘(a (b c (a) a) (a b)))
   returns ((b c ( )) (b))

7. The Scheme function reverse described in the notes reverses only the “top level” of a list: if L = ((2 3) 4 (5 6)) then (reverse ‘L) = ((5 6) 4 (2 3)). Write a Scheme function deep-reverse that also reverses all sublists (deep-reverse L) - ((6 5) 4 (3 2)).

8. Write a Scheme function with three parameters, two atoms and a list, that returns the list with all occurrences, no matter how deep, of the first atom replaced by the second atom. For example:
   (replaceatom ‘a ‘x ‘(a (b c (a) a) (a b)))
   returns (x (b c (x) x) (x b)))

9. What will the following return?
   (map fact '(2 3 4))

10. What will the following return?
   (map (lambda (x) (- 0 x)) '(-1 2 3 -4 5))

11. Assume we define the following function:
   (define (is-negative? x) (< x 0))

   What will the following return?
   (map is-negative? '(-1 2 3 -4 5))

12. Write a function remove-if that will remove elements of a list which meet the conditions expressed in a predicate f (i.e. remove the elements for which the function f is true).
   Examples:
   (remove-if even? '(7 8 12 13 15)) will return (7 13 15)
   (define (remove-if f L)

13. What will the following return?
   (remove-if (lambda (x) (< x 0)) '(-1 2 3 -4 5))

14. Write a tail recursive version of remove-if.

15. Define a function add-n with an integer n as a parameter which returns a function which will add n to its parameter.
   Examples:
   ((add-n 5) 6) will return 11
   (define (add-n n)

16. Use the function you have just defined to define a function add-6 which will add 6 to its parameter.
   (define add-6

17. What would be displayed by (display (add-6 8))

18. Using remBuilder define a function remAtom which will remove the top atoms from a list.