;;; Q1. (numcount L) : return count of numbers in List L

;;; Base Case : List is empty, return 0

;;; Assumption : (numcount M) will return count of numbers in any list M smaller than L

;;; Step : If (car L) is a list then return count of numbers in (car L) + count of numbers in (cdr L)

;;; If (car L) is a number then return 1 + count of numbers in (cdr L)

;;; Else return count of numbers in (cdr L)

(define (numcount L)

(cond ((null? L) 0)

((list? (car L)) (+ (numcount (car L)) (numcount (cdr L))))

((number? (car L)) (+ 1 (numcount (cdr L))))

(else (numcount (cdr L)))

))

;;; Q2. (insert x L) : Insert x in sorted list L

;;;Base Case : List is empty, return x as a List

;;;Assumption : (Insert x M) will insert x in sorted list M smaller than L

;;;Step : If x > (car L) then return combined list of (car L) and x inserted in (cdr L)

;;; Else return combined list of x as first element followed by all elements in L

(define (insert x L)

(cond ((null? L) (cons x '()))

((> x (car L) ) (cons (car L) (insert x (cdr L))))

(else (cons x L))))