SIXTH QUIZ

You have 15 minutes from the start of class to complete this quiz. Read the questions with care; work with deliberate speed. Don't give us more than we ask for. The usual instructions apply. Good luck!

Problem 1 (6 points)

What is the value of each of the following expressions? (Remember that `even?` returns true if its input is an even number.)

```
(define pair?
  (lambda (a-list)
    (= 2 (length a-list))))

(define L '( (3 4)  (5 6)  (1 2 3)  (7 8)  ()  (9 10 11) (2) ))
```

(a) `(filter even? (list 17 12 14 13 11))`

(b) `(filter pair? L)`

(c) `(map add1 (list 5 6 7 8))`

(d) `(map (lambda (N) (* N N)) (list 2 3 4 5))`

(e) `(foldr * 1 (list 2 5 8))`

(f) `(foldr (lambda (L1 L2) (append L1 L2)) empty L)`

(g)
```
(filter (lambda (n) (> n 7))
        (map (lambda (LON) (foldr + 0 LON)) L))
```

Problem 2 (19 points)

Suppose we have a list called BL of books defined as follows:

```
(define-struct book (title author genre price sold instock))
```

where title and author are strings, genre is a symbol (e.g., `'cookbook` or `'humor`) representing the category of the book, price is a number representing the price of one copy, sold is the number of copies sold, and instock is the number of copies in stock.

(a) (9 points) For each of the following expressions, describe in one clear and precise English phrase what value it returns. Don't just say, "It does a foldr of plus and zero and ..."; give a description of what the expression *means,* something you could put in a software catalog so that a prospective buyer could find what he or she wanted. Say something like, "a list of the authors whose books earned over $1,000,000."

(a.1)
```
(map book-title (filter (lambda (B) (= 0 (book-instock B))) BL))
```

(a.2)
```
(local ((define chosen (filter (lambda (B) (equal? (book-genre B) 'fiction)) BL)))
  (/ (foldr + 0 (map book-price chosen))
     (length chosen)))
```

(a.3)
```
(foldr + 0 (map (lambda (B) (* (book-price B) (book-instock B)))
                (filter (lambda (B) (or (equal? (book-genre B) 'drama)
                                        (equal? (book-genre B) 'poetry))) BL)))
```

(b) (5 points) Using `map`, `filter`, and/or `foldr`, define the following function without using explicit recursion.

```
;; high-earning-books: number list-of-books -> list-of-books
;; Return a list of those books on the input list that have earned more than
;; the specified (input) number.
(define high-earning-books
  (lambda (cutoff booklist)
```

(c) (5 points) Using `map`, `filter`, and/or `foldr`, define the following function without using explicit recursion. Use previously defined functions where possible.

```
;; high-earning-authors: number list-of-books -> list-of-string
;; Return a list of the names of the authors of books on the input list
;; that have earned more the specified (input) number.
(define high-earning-authors
  (lambda (cutoff booklist)
```