Informatics 41 • Fall 2011 • David G. Kay • UC Irvine • Your name ______________________________

SEVENTH QUIZ

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

Brief reminders: `zero?` returns true if its input is zero; `negative?` returns true if its input is less than zero; `sub1` subtracts one from its argument; `add1` adds one to its argument.

Problem 1 (5 points)

What is the value of each of the following expressions?

(a) `(map sub1 (list 5 4 3 2 1))`

(b) `(map (lambda (s) (string-append “Mc” s)) (list “Donald’s” “Carthy” “Duck”))`

(c) `(filter negative? (list 12 -11 (* 5 -2) (/ 250 10) (sub1 -20)))`

(d) `(filter (lambda (n) (not (zero? n))) (list 0 (- 7 3) (- 8 8) (* 5 0) 14))`

(e) `(foldr + 0 (list 10 20 30 40))`

(f) `(build-list 6 (lambda (n) n))`

(g) `(build-list 4 add1)`

Problem 2 (7 points)

A binary search tree is either empty or a node defined as `(define-struct node (rootvalue leftsubtree rightsubtree))`. For this problem, each root value will be a single string. Complete the following definition by filling in the blanks, one function, constant, or other identifier per blank:

```
;; BST-insert:  string  BST  ->  BST
;; Return the BST with the input string inserted in the appropriate place.
(define BST-insert
  (lambda (s T)
    (cond
      ((empty? T) (make-node ________________ ________________ ________________))

      ((string=? s (________________ T)) ________________) ; Just ignore duplicates

      ((string<? s (________________ T))

        (make-node (________________ T)

                   (________________ s (________________ T))

                   (________________ T)))

      (else (make-node (________________ T)

                       (________________ T)

                       (________________ s (________________ T)))))))
```

Problem 3 (8 points)

Suppose that the Anteater Bookstore represents its inventory as a list of book structures defined as follows:

```
     (define-struct book (author title ISBN category price instock))
```

where author, title, ISBN, and category are strings, and price and instock are numbers. ISBN is the International Standard Book Number and instock is the number of copies currently in stock.

Define the function below, according to its contract and purpose statement. You may do it conventionally (using `cond` and explicit recursion) or using `map/filter/foldr` (without explicit recursion). You don't need anywhere near the amount of blank space on this page.

```
;; author-category-in-stock:  list-of-book  string(author)  string(category) -> number
;; Return the number of books in stock with the specified author and category
;; Example:  (author-category-in-stock AnteaterBooks "Donald Knuth" "Computers")
;; returns the total number of copies (in stock in the AnteaterBooks store)
;; of books on computers by Donald Knuth.
(define author-category-in-stock
  (lambda (L a c)
```