Informatics 41 • Fall 2010 • David G. Kay • UC Irvine • Your name ______________________________

SIXTH 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!

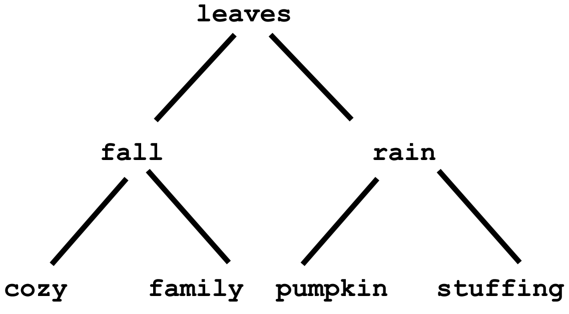
Problem 1 (5 points)

A book is a structure with five fields: strings for the title, author, and genre (the type of book, like cookbook or mystery) and numbers for the price and number of copies in stock. We define the structure with `(define-struct book (title author genre price instock))`. Fill in the blanks to complete the function described below; each blank requires exactly one symbol, function name, or constant.

```
;; computer-book?:  book  ->  boolean
;; Return true if the book's genre is "computer"
(define computer-book?
  (lambda (B)
    (string=? (book-genre B) "computer")))
;; computer-books-in-stock:  list-of-book  ->  number
;; Return the total number of computer books in stock.
(define computer-books-in-stock
  (lambda (L)
    (cond
      ((________________ L) ________________)

      ((________________ (________________ L))

        (________________ (________________ (________________ L))

                          (________________ (________________ L))))

      (else (________________ (________________ L))))))
```

Problem 2 (5 points)

(a) (1 point) At the right is a picture of a binary search tree with the items ordered alphabetically. Insert the value "fireplace" into the tree; draw a new branch and node to indicate where it belongs. Be careful to distinguish a left subtree from a right subtree, if necessary (by the angle of the branch).



(b) (1 point) Now insert the value "turkey" into the tree.

(c) (2 points) List all nine items in the tree in the order they would be visited in an inorder traversal of the tree. In other words, if you converted this BST to a list using an inorder traversal, what would be the order of items in the list?

(d) (1/2 point) In a *preorder* traversal of the original tree (without "fireplace" or "turkey"), what is the very *first* node processed?

(e) (1/2 point) In a *postorder* traversal of the of the original tree , what is the very *last* node processed?

Problem 3 (10 points)

(a) (9 points) Suppose we have a binary search tree of book structures (defined as above), ordered alphabetically by the books' titles, with nodes defined as `(define-struct node (value left right))`, where `value` is a book and `left` and `right` are either empty or a node (i.e., a subtree), and the binary search tree property holds. Complete the definition below, adding the necessary code in the blank spaces.

```
;; book-title-less?:  book  book  ->  boolean
;; Return true if first book's title is alphabetically less than the second
(define book-title-less?
  (lambda (B1 B2)
    (string<? (book-title B1) (book-title B2))))

;; book-title-greater?:  book  book  ->  boolean
;; Return true if first book's title is alphabetically greater than the second
(define book-title-greater?
  (lambda (B1 B2)
    (string>? (book-title B1) (book-title B2))))

;; add-new-book:  book   BST-of-books  ->   BST-of-books
;; Insert the input book into the input BST according to the alphabetical
;; value of the book's name; return the new tree
(define add-new-book
  (lambda (B T)
    (cond
      ((empty? T) (make-node


        ))
      ((book-title-less? B (node-value T)) (make-node






        ))
      ((book-title-greater? B (node-value T)) (make-node






        ))
      (else T))))
```

(b) (1 point) How does this code handle two books with the same name?