Informatics 41 • Fall 2009 • David G. Kay • UC Irvine • Your name ______________________________

FIFTH 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!

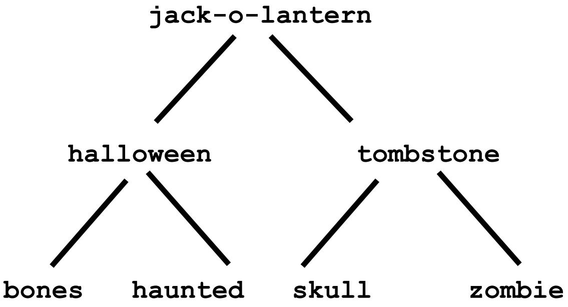
Problem 1 (4 points)

Fill in the blanks in the function described below. All the parentheses are in the correct places and each blank should be filled by exactly one item: function name, constant, or other name.

```
(define-struct rrant (name cuisine phone dish price))
;; collect-rrant-names:  list-of-rrant -> list-of-string
;; Return a list containing just the names of the rrants on the input list.
(define collect-rrant-names
  (lambda (L)
    (cond

      ((_______________ L) _______________)

      (else (________________ (________________ (________________ L))

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

Problem 2 (6 points)

(a) (1 point) At the right is a picture of a binary search tree with the items ordered alphabetically. Insert the value "ghost" 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 "skeleton" 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 point) In a *preorder* traversal of the original tree (without "ghost" or "skeleton"), what is the very *first* node visited? ___________________ What is the *second* node visited? ___________________

(e) (1 point) In a *postorder* traversal of the of the original tree (without "ghost" or "skeleton"), what is the very *first* node visited? ___________________ What is the *second* node visited? ___________________

Problem 3 (8 points)

Suppose we have a binary search tree of `rrant` structures (defined as above), with nodes defined as follows:

```
(define-struct node (root left right))
```

where `root` is a `rrant`, `left` and `right` are either empty or a node, and the binary search tree property applies. Complete the definition below of `named-rrant-in-BST?`, adding the necessary code in the blank spaces.

```
;; named-rrant-in-BST?:  string  BST-of-rrants  ->  boolean
;; If the string matches the name of a rrant in the tree, return true.
;; Otherwise, return false.
(define named-rrant-in-BST?
  (lambda (n T)
    (cond
      ((empty?



      ((string=?



      ((string<?



      (else
```

Problem 4 (2 points)

True or false (and if it's false, say very briefly how you know it's false):

(a) There is only one correct way (one correct algorithm) to perform any given computational task.

(b) Any algorithm that performs a given task will have the same performance (i.e., will do that task on a given set of data in about the same amount of time as other algorithms that do the same task).