NINTH 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 (13 points)

Along with this quiz is a copy of the Restaurants program.

(a) (1 points) What data structure does this program use to implement a single restaurant: A structure, a list, a vector, a binary search tree, or something else?

(b) (1 point) What data structure does this program use to implement the collection of restaurants: A structure, a list, a vector, a binary search tree, or something else?

(c) (2 points) One or more of the collection functions are recursive: List their names. [Don't consider the view/controller functions or the restaurant/menu/dish functions.]

(d) (2 points) One or more of the collection functions are tail-recursive: List their names.

(e) (2 points) Pick one of the recursive collection functions that isn't tail-recursive. Copy below the line(s) of code from that function that make it not tail-recursive.

(f) (2 points) One or more of the collection functions uses the accumulator style: List their names.

(g) (2 points) Why do the collection functions include `collection->list`? What purpose does it serve?

(h) (1 point) If you ignore the collection functions whose names end in `-alternative` or `-original` or `-aux`, you see that the search, remove, and change functions all depend on `collection-process`. We have seen examples before of collecting similar functionality together. Why did the author of this code generalize all these operations into one place; what was he trying to avoid? [Two words are enough.]

Problem 2 (7 points)

(a) (1 point) What's good about redundancy?

(b) (1 point) What's bad about redundancy?

(c) (1 point) What does data compression do with redundancy?

(d) (2 points) What's the difference between lossless and lossy compression?

(e) (2 points) Why would anyone use lossy compression?