Informatics 41 • Fall 2005 • David G. Kay • UC Irvine   Your name ______________________________

SEVENTH QUIZ

Your student ID ______________________________

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 (9 points)

Complete the definition of the function `add-book-info`, which uses book structures like this:

```
(define-struct book (author title genre price sales))

;; add-book-info: book (listof book) -> (listof book)
;; If the book (the first argument) is already on the list (i.e., its author and title
;; match an existing book), update the book's entry using update-book (below).
;; If the book is new, just add it to the list.
(define add-book-info
  (lambda (new-book booklist)
    (cond
    ((empty? booklist) (list ____________))

    ((and (string=? (book-author ____________) (book-author (____________ ____________)))

          (string=? (book-title ____________) (book-title (____________ ____________))))

       (cons (____________ ____________ (____________ ____________)) (rest booklist)))

    (else (cons (first booklist) (____________ new-book (rest booklist)))))))


(define update-book
  (lambda (new-book existing-book)
    (make-book (book-author existing-book) (book-title existing-book)
               (book-genre existing-book) (book-price existing-book)
               (+ (book-sales new-book) (book-sales existing-book)))))
```

Problem 2 (4 points)

Here is a definition of `collection-search` from the restaurant collection program; it could be used just as well to search a collection of books; we've just changed "rrant" to "book" in the contract and purpose:

```
;; collection-search: collection (book->boolean) -> collection
;; Return a collection made up of all the books in C that pass test?
(define collection-search
  (lambda (C test?)
    (cond
      ((collection-empty? C) (make-collection 'x))
      ((test? (first C))
       (cons (first C) (collection-search (rest C) test?)))
      (else (collection-search (rest C) test?)))))
```

[continued on the other side]

Complete the definition below by including an appropriate lambda expression::

```
;; search-by-genre: collection -> collection
;; Return a collection of those books in the input collection
;; that match the genre specified by the user.
(define search-by-genre
  (lambda (BL)
    (local ((define genre (begin
                            (display “Please enter genre to search for:  “)
                            (read))))
      (collection-search
        BL






))))
```

Problem 3 (7 points)

A Deus X reference sheet is attached. You may tear it off; you don't have to turn it back in. If you show your work (e.g., draw a picture of the register(s) and/or memory locations), we'll have a better chance of assigning you partial credit.

(a) (3 points) Suppose that location 5555 of the Deus X machine's memory holds the number 25 and that location 3333 holds the number 20. What is in location 3333 after executing these three instructions? (The first number on each line indicates the instruction's address in memory.)

```
0.   10  5555  (lda 5555)
1.    2  3333  (sub 3333)
2.   20  3333  (sta 3333)
```

(b) (4 points) Suppose that location 7777 of the Deus X machine's memory holds the number 20, that location 8888 also holds the number 20, and location 9999 holds the number 65. What does the Deus X machine print after executing these instructions?

```
0.   10  7777  (lda 7777)
1.   50  8888  (cmpa 8888)
2.   60  5     (je 5)
3.    6  7777  (out 7777)
4.    7  6     (jmp 6)
5.    6  9999  (out 9999)
6.  …
```