

## THIRD 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!

You may show lists in any of three ways: `(cons 'AC (cons 'DC empty))`, `(list 'AC 'DC)`, or `'(AC DC)`.

### Problem 1 (2 points)

Evaluate each of the following expressions. (That is, what does DrScheme display in the interactions window when you enter the expression or click Run?)

```
(define ELECTORS 538)
(define NEED-TO-WIN (+ 1 (/ ELECTORS 2))) ; 270 needed to win, 50% + 1
```

(a) `(>= ELECTORS 535)`

(b) `(and (>= NEED-TO-WIN 270)
 (< NEED-TO-WIN ELECTORS))`

(c) `(+ 10
 (cond
 ((= ELECTORS (+ 435 100 3)) 55)
 (else 44 )))`

### Problem 2 (5 points)

Evaluate each of the following expressions. Use this definition independently for each of the five parts:

```
(define L (cons 'AA (cons 'BB (cons 'CC (cons 'DD empty)))))
```

(a) `(first L)`

(b) `(rest L)`

(c) `(first (rest L))`

(d) `(cons 'RN (rest (rest L)))`

(e) `(cond
 ((equal? (cons 'OK empty) 'OK) 'orange)
 ((empty? (rest (rest (rest L)))) 'grapefruit)
 ((equal? (first (list 'OK)) 'OK) 'lime)
 (else 'lemon))`

**Problem 3** (11 points)

We define a book as a structure

```
(define-struct book (author title year wholesale retail))
```

where `author` and `title` are strings and `year`, `wholesale`, and `retail` are numbers (representing the year the book was published, the book's wholesale price, and its retail price).

(a) (3 points) Define the function `book-profit-margin` as described below.

```
;; book-profit-margin: book -> number
;; Return the difference between the retail price and the wholesale price of a book
```

(b) (8 points) Define the function `increase-profit` as described below. Where applicable, use functions you have already defined rather than duplicating code.

```
;; increase-profit: book number -> book
;; If the input book's profit margin is at least the specified number, return the
;; input book unchanged. Otherwise, return the input book with its retail price
;; increased by the input number.

(define increase-profit
  (lambda (B threshold)
    (let ([profit (- (book-retail B) (book-wholesale B))])
      (if (>= profit threshold)
          B
          (make-book (book-author B) (book-title B) (book-year B) (book-wholesale B) (+ (book-retail B) threshold))))))
```

```
(check-expect (increase-profit (make-book "X" "Y" 2008 25.00 35.00) 10.00)
               (make-book "X" "Y" 2008 25.00 35.00))
(check-expect (increase-profit (make-book "X" "Y" 2008 25.00 35.00) 20.00)
               (make-book "X" "Y" 2008 25.00 55.00))
```

**Problem 4** (2 points)

Look back at the top of Problem 1. In just a few English words, why did we define `NEED-TO-WIN` as we did, instead of just `(define NEED-TO-WIN 270)`?