

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!

You may show lists in any of three ways: `(list 73 15)`, `'(73 15)`, or `(cons 73 (cons 15 empty))`.

Problem 1 (4 points)

Complete the definition of the function below according to the contract and purpose statement shown, with one function or parameter name in each blank. The predefined function `zero?` returns true if its argument is zero; the predefined function `add1` adds 1 to its argument.

```
;; count-zeroes: list-of-numbers -> number
;; Return the number of zeroes on the input list.
(check-expect (count-zeroes empty) 0)
(check-expect (count-zeroes (list 1 0 2 0 3 0 0)) 4)

(define count-zeroes
  (lambda (L)
    (cond
      ((_____ L) _____)

      ((zero? (_____ L)) (_____ (_____ (_____ L))))

      (else (_____ (_____ L))))))
```

Problem 2 (5 points)

Complete the definition of the function below according to the contract, purpose, and examples shown. [Hint: The solution requires `cond` plus two more lines.]

```
;; halve-all: list-of-numbers -> list-of-numbers
;; Return the input list with each value divided by two
(check-expect (halve-all empty) empty)
(check-expect (halve-all (list 2 4 6 8 10 11)) (list 1 2 3 4 5 5.5))

(define halve-all
  (lambda (L)
```

Problem 3 (9 points)

(a) (3 points) A portrait-style image is taller than it is wide. A landscape-style image is wider than it is tall. Complete the definition of the function below according to its contract and purpose statement. You will want to use the predefined functions `image-height` and `image-width`, each of which takes an image and returns a number.

```
;; portrait?: image -> boolean
;; Return true if the input image is taller than it is wide
(define portrait?
  (lambda (I)
```

(b) (6 points) Complete the definition of the function below according to the contract and purpose statement shown, with one function or parameter name in each blank. Where applicable, use functions you have already defined rather than duplicating code.

```
;; keep-portraits: list-of-images -> list-of-images
;; Return a list of the portrait-oriented images on the input list
(define keep-portraits
  (lambda (L)
    (cond
      ((_____ L) _____)

      ((_____ (_____ L))

        (_____ (_____ L) (_____ (_____ L))))

      (else (_____ (_____ L))))))
```

Problem 4 (2 points)

On this quiz there are four function definitions. One function does a mapping operation, one does a filtering operation, one does a folding/reducing operation, and one does none of those. Which is which?

Mapping function: _____

Filtering function: _____

Folding/reducing function: _____

None of those: _____