CS 61A Midterm #1 — February 17, 2010

Your name _____

login: cs61a–_____

Discussion section number _____

TA's name _____

This exam is worth 40 points, or about 13% of your total course grade. The exam contains seven substantive questions, plus the following:

Question 0 (1 point): Fill out this front page correctly and put your name and login correctly at the top of each of the following pages.

This booklet contains six numbered pages including the cover page. Put all answers on these pages, please; don't hand in stray pieces of paper. This is an open book exam.

When writing procedures, don't put in error checks. Assume that you will be given arguments of the correct type.

Our expectation is that many of you will not complete one or two of these questions. If you find one question especially difficult, leave it for later; start with the ones you find easier.

If you want to use procedures defined in the book or reader as part of your solution to a programming problem, you must cite the page number on which it is defined so we know what you think it does.

READ AND SIGN THIS:

I certify that my answers to this exam are all my own work, and that I have not discussed the exam questions or answers with anyone prior to taking this exam.

If I am taking this exam early, I certify that I shall not discuss the exam questions or answers with anyone until after the scheduled exam time.

_____

0	/1
1	/5
2-3	/10
4	/4
5	/4
6	/8
7	/8
total	/40

Your name _____ login cs61a–_____

Question 1 (5 points):

What will Scheme print in response to the following expressions? If the expression produces an error message, you may just write "error"; you don't have to provide the exact text of the message. If the value of an expression is a procedure, just write "procedure"; you don't have to show the form in which Scheme prints procedures.

```
> (keep (lambda (wd) (= (count wd) 3)) '(for no one))
```

```
> (define (f x) (lambda () (se 'hello x)))
> (first (f 'brian))
```

```
> (define (g x y) (x (se 'hello y)))
> (g first 'satish)
```

```
> (every (* 2) '(1 2 3 4))
```

```
> (cadr '((a b c) d e f))
```

Question 2 (6 points):

(a) What will Scheme print in response to the following expression? Also draw a box and pointer diagram for the value produced by the expression. Don't forget the start arrow!

```
> (cons (cons 1 2) (cons 3 '()))
```

(b) Draw the **box and pointer diagram** for the list `(a () (b c))`. Then, using only quoted symbols (words), the empty list, and the `cons` procedure, give an expression that creates the list.

Question 3 (4 points):

(a) Assume `foo` is a *list* of *sentences*. Write an expression that returns the final *letter* of the first *word* in the sentence at the beginning of the list `foo`.

(b) Assume `bar` is a *pair* of *2d-points*, i.e., points in the x-y plane. Write an expression to compute the slope of the line through the points. You must use the data abstraction *2d-points* with constructor **make-point** and selectors **x-coord** and **y-coord**.

Question 4 (4 points):

What is the order of growth of each of the following procedures?

```
(define (append a b) ; helper function for INTEGERS-TO
  (if (null? a)
      b
      (cons (car a) (append (cdr a) b))))

(define (integers-to n)
  (if (= n 0)
      '()
      (append (integers-to (- n 1))
              (list n) )))
```

The running time of `integers-to` is Θ(_____)

```
(define (mystery n)
  (cond ((= n 0) 0)
        ((= n 1) 1)
        (else (+ (mystery (- n 1))
                 (mystery (/ n n))))))
```

The running time of `mystery` is Θ(_____)

Question 5 (4 points):

Do the following procedures generate an iterative or recursive process? Place a check mark next to the appropriate answer.

```
(define (slow-multiply a b)
   (define (slow-multiply-internal a b result)
      (cond ((= a 0) result)
            (else (slow-multiply-internal (- a 1) b (+ result b)))))
   (slow-multiply-internal a b 0))
```

Iterative _____ Recursive _____

```
(define (faster-multiply a b)
   (cond ((= a 0) 0)
         ((odd? a) (+ (faster-multiply (quotient a 2) (* b 2)) b))
         (else (faster-multiply (quotient a 2) (* b 2)))))
```

Iterative _____ Recursive _____

Your name _____ login cs61a–_____

Question 6 (8 points):

Write a procedure `vowel-counts` that takes a sentence and returns a sentence of the number of vowels in each word. (The CS61A TAs like to refer to this as "evoweluation".) You can assume the procedure `vowel?` already exists.

```
(define (vowel? letter) (member? letter 'aeiou))
```

Your solution should use only **higher-order procedures (no recursion)**!

```
> (vowel-counts '(consonants are better))
(3 2 2)

(define (vowel-counts sent)
```

Your name _____ login cs61a–_____

Question 7 (8 points)

Write a procedure `contains-phrase?` that takes two sentences. It should return `#t` if the first sentence contains the second sentence in order (but not necessarily consecutively).

```
> (contains-phrase? '(know if cs61a is a fun class?) '(cs61a is fun))
#t
> (contains-phrase? '(students like turtles) '(turtles like students)
#f
> (contains-phrase? '(let it be) '(let it be))
#t
```

```
(define (contains-phrase? sent phrase)
```