CS 61A Midterm #1 — February 7, 1994

Your name _____

login cs61a–_____

Discussion section number _____

TA's name _____

This exam is worth 20 points, or about 11.5% of your total course grade. The exam contains five 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.

CS 3 alumni please note: Don't use the CS-3-only higher order functions (every, keep, accumulate) in these problems!!

0	/1
1	/3
2	/1
3	/5
4	/5
5	/5
total	/20

Question 1 (3 points):

What will Scheme print in response to the following expressions? Assume that they are typed in sequence, so definitions affect later interactions. If an expression produces an error message, you may just say "error"; you don't have to provide the exact text of the message. If the value of an expression is a procedure, just say "procedure"; you don't have to show the form in which Scheme prints procedures.

```
(+ (* 3 4 0 7) 8)
```

```
(lambda (x) (/ x 0))
```

```
(butfirst '(help!))
```

```
((if 3 * +) 4 5)
```

```
(let ((+ -))
  (+ 8 2))
```

```
(+ 9 3)
```

Your name _____ login cs61a–_____

Question 2 (1 point):

One or more of the following procedures generates an iterative process. Circle them. Don't circle the ones that generate a recursive process.

```
(define (butfirst-n num stuff)
  (if (= num 0)
      stuff
      (butfirst-n (- num 1) (bf stuff))))
```

```
(define (member? thing stuff)
  (cond ((empty? stuff) #f)
        ((equal? thing (first stuff)) #t)
        (else (member? thing (bf stuff)))))
```

```
(define (addup nums)
  (if (empty? nums)
      0
      (+ (first nums)
         (addup (bf nums)))))
```

Question 3 (5 points):

Write a predicate procedure `increasing?` that takes as its argument a sentence of numbers. It should return true if all the numbers are in increasing order (not allowing equality), false otherwise:

```
> (increasing? '(4 17 25 32))
#t

> (increasing? '(4 17 3 32))
#f

> (increasing? '(4 17 17 32))
#f

> (increasing? '(4))
#t
```

Assume that the argument sentence will always contain at least one number.

Your name _____ login cs61a–_____

Question 4 (5 points):

A strategy procedure in the 21 project takes two arguments, the player's hand so far and the dealer's visible card, as in this example:

```
(my-strategy '(3h 4d 10c) '5s)
```

It returns true to indicate that the player wants another card, false otherwise.

(a) Write a strategy `four-cards` that always takes another card if the player has fewer than four cards, but stops when the player has four cards. (This isn't a very good strategy!) Reminder: you can use `count` to count the number of words in a sentence.

(b) Write a procedure `n-cards` that takes a number `n` as its argument, returning a strategy that stops when the player has `n` cards. For example, `(n-cards 4)` should return a strategy equivalent to the one you wrote in part (a).

Question 5 (5 points):

Some people from Brooklyn interchange the sounds of "oi" and "er." For example, they pronounce "fern" as "foin" and "foil" as "ferl."

Write a procedure `brooklyn` that takes a word as its argument, returning a word in which each `oi` is replaced by `er` and each `er` is replaced by `oi`:

```
> (brooklyn 'mermaid)
MOIMAID

> (brooklyn 'hoi-polloi)
HER-POLLER

> (brooklyn 'salami)
SALAMI
```

Assume that the argument word will always contain at least one letter.