CS 61A Midterm #2 ver1.03 -- March 2, 1998 **Exam version: A**

Your name _____

login: cs61a-_____

Discussion section number _____

TA's name _____

Look at the edge of your seat. Write your ROOM, seat row and number. Your row number may not be visible form where you sit, so we will help you later. _____

This booklet contains 6 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, worth 40 points, or about 13% of your total course grade. The exam contains 6 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.

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.

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 at other than the regularly scheduled time, I certify that I have not/ will not discuss the exam questions or answers with anyone inappropriately.

_____

0

/1 1 /6 2 /6 3 /5 4 /6 5 /10 6 /6 total /40

1

Your name _____ login cs61a-___**Exam version: A**

Question 1 (6 points): Draw box-and-pointer diagrams that represent the results of the following expressions. (1 point each)

```
(define x1 (list (cons 'a 'b) (list 'c 'd)))
```

```
(define x2 (cons 'a (cons 'b (cons 'c 'd))))
```

```
(define x3 (list 'a 'b 'c (cons 'd '())))
```

```
(define x4 (append (list (cons 'a 'b)) (list '())))
```

```
(define x5 '((1 2) 3 . 4))
```

```
(define x6 '(1 (2 . 3)))
```

Question 2 (6 points): Assume that we are using a representation of a matrix as a list of lists, namely `((1 2 3) (4 5 6) (7 8 9))` is the matrix H:

```
1 2 3
4 5 6
7 8 9
```

A. Given this definition `(define (chop m)(map cdr (cdr m)))`, what does `(chop H)` return?

B. Given this definition `(define (foo m) (if (empty? M) '() (cons m (foo (chop m)))))`, what does `(foo H)` return?

C. Given the definition of accumulate on p. 116, what does this return?

```
(accumulate * 2 (map caar (foo H)))
```

[note for reference, here are definitions of accumulate and caar:

```
(define (accumulate op init seq)
  (if (null? seq) init (op (car seq) (accumulate op init (cdr seq)))))
```

```
(define (caar x) (car (car x)))]
```

2

Your name _____ login cs61a-___**Exam version: A**

Question 3 (5 points):

```
(define (triad left label right)
  (lambda (m)
    (cond ((eq? m 't-car) left)
          ((eq? m 't-cbr) label)
          ((eq? m 't-cdr) right))))
```

The above constructor implements a triad [left label right] through message passing. Triads can be used to create binary trees; each node has a label, a left child, and a right child. Leaf nodes will have empty lists as their left and right children.

Define selectors that retrieve the label of the current node, and the left and right child branches.

```
(define (label node) _____ )



(define (left node) _____ )



(define (right node) _____ )
```

```
      5
    /   \
   3     8
  / \   / \
 1   4 7   11
```
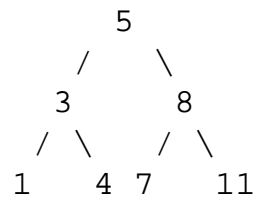
Write the Scheme code to create the above tree with the triad constructor. Make sure that your selectors will work with your tree representation.

3

Your name _____ login cs61a-___**Exam version: A**

Question 4 (7 points):

A multi-set is represented as a list of elements, some of which may be repeated. For example, (define S '(2 1 2 3 4 2)) creates a multi-set, S, with six elements having four unique values, since there are three repetitions of 2. Another way to represent multi-sets would be to explicitly keep count of repeated elements, creating cons pairs of (*value . repetitions*), in which case case S might be ((2 . 3) (1 . 1) (3 . 1) (4 . 1)). The ordering of the pairs does not matter.

You *must* use the following functions to solve this problem, but you'll have to figure out what they do first.

```
(define (count-if-member x s)
 (if (null? S)
     0
     (+ (count-if-member x (cdr s))
        (if (equal? (car s) x) 1 0))))
```

A. In one English sentence, describe what count-if-member does. (1 pt)

```
   (define (remove x s)
    (cond ((null? S) '())
          ((equal? (car s) x) (remove x (cdr s)))
          (else (cons (car s) (remove x (cdr s))))))
```

B. In one English sentence, describe what remove does. (1 pt)

Complete the program below that transforms multi-sets in the first representation to the second representation (4 pts).

C. Write the program `transform-multiset`. If you did it the way we expect you to, you can identify each space like `-A-` in the skeleton function below with the function you used in your version. Look for the function name in the list below, and write the number next to the letter of the space it fits in. You may use a number more than once. If no numbered function fits, write out the name in the space below.

```
(define (transform-multiset s)
  (if (-A- s)
      '()
      (-B- (-C- (-D- s) (-E- (-F- s) s))
           (-G- (-H- (-I- s) (-J- s)))))))
```

Here are your choices

```
1 car              4 cdr        7 union              10 map          13 equal?
2 cadr             5 cons       8 transform-multiset 11 accumulate   14 append
3 count-if-member  6 empty?     9 if                 12 remove       15 list
```

Here is where you fill in the blanks. We did one to show you how:

A:_____ B:_____ C:_____ D:_____ E:_____ F:_____ G__8__H:_____ I:_____ J:_____

Your name _____ login cs61a-___**Exam version: A**

Question 5 (10 points): Here is a set of definitions of data objects, using message passing

```
(define (make-inches L)
  (lambda (unit)
    (cond ((eq? unit 'inches) L)
          ((eq? unit 'feet) (/ L 12))
          ((eq? unit 'centimeters) (* L 2.54))
          (else (error "Unknown unit")))))
```

```
(define (make-feet L)
  (lambda (unit)
    (cond ((eq? unit 'inches) (* L 12))
          ((eq? unit 'feet) L)
          ((eq? unit 'centimeters) (* L 30.48))
          (else (error "Unknown unit")))))
```

```
(define (make-centimeters L)
  (lambda (unit)
    (cond ((eq? unit 'inches) (/ L 2.54))
          ((eq? unit 'feet) (/ L 30.48))
          ((eq? unit 'centimeters) L)
          (else (error "Unknown unit")))))
```

The above functions are constructors for a length object.

   A. Write a selctor "length-value" that takes a length object and the units to be used for the result, and returns the
       numerical value of the length. (2 pts)

B. Write a procedure called "area-of-rectangle-in-sq-feet" that takes two arguments: two length objects which represent the two sides of a rectangle and returns the result in square feet. (2 pts)

C. A numerical package could use the same approach to multiplication: coerce all types to integers, multiply the integers and then coerce the result back to the relevant type. In one English sentence explain why this is a bad idea. (1 pt)

D. In one or two English sentences, defend or attack the argument that coercion to any type (float or rational) is also bad. (2 pts)

E. If we write a numerical package with three types of number: integer, rational and float explain why we do not need 6 type coercion routines. What is the minimum number we do need? (3 pts)

5

Your name _____ login cs61a-___**Exam version: A**

Question 6 (6 points):

True or false?

_____`(car ''(hello))` is `(quote)`.

_____According to the definition of interval on page 93-94, for any interval I, the result of computing a mathematical function $f(I)$ should represent the range of the result. If I is the interval from 0 to Π and f is the sine

function in radians, and sin(0) = sin(*Π*) = 0 then *f(I)* is the interval from 0 to 0.

_____car has the closure property of section 2.2. page 97.

_____`(filter (lambda (x) (cons x x)) y)` returns a list of pairs. (See page 115 for definition of filter if you don't recall it.)

_____`make-rat` on page 88, and `attach-tag` on page 176 are the same abstraction because they are both the same as cons.

_____In your project, frames are described by three vectors. If two frames of the same size are at the exact same position, each of the three vectors must be the same.

6