

91.301, Organization of Programming Language
Fall 2004, Prof. Yanco

Sample Quiz 1

Problem 1 (9 points)

What is the result of the evaluation of the final statement in each of the following groups of Scheme expressions? Assume that each group is evaluated separately.

```
(define a 1)
```

```
(let ((a 4)
      (b (+ a 2)))
      (* a b))
```

```
(define (f x) (x 3 4))
```

```
(f (lambda (y z) (+ (square y) z)))
```

```
(define (t f)
  (lambda (x) (f (f (f x)))))
```

```
(define (add2 x)
  (+ x 2))
```

```
((t add2) 0)
```

Problem 2 (15 points)

Assume the following Scheme expressions are evaluated sequentially. Fill in the table below, using `eq?`, `eqv?` and `equal?` to compare the items in the first two columns.

```
(define a (list 1 (list 'a 'b) 2))
```

```
(define b (list 1 (list 'a 'b) 2))
```

```
(define c (list 1 a))
```

		eq?	eqv?	equal?
a	b			
(cadr a)	(cadr b)			
(caadr a)	(caadr b)			
(car b)	(car c)			
(cdr a)	(cdadr c)			

Problem 3 (16 points)

- a. Draw the box and pointer diagrams for the following lists.
- b. Write the sequence of cars and cdrs needed to get the number 3 out of each list.

```
(list 1 (list 2 3) (list 4))
```

```
(cons (list 1 2) (cons 3 4))
```

Problem 4 (20 points)

The athletics department has asked us to help them write code that will track the performance of the athletic teams. Each team record will contain the name of the team, the number of wins, the number of ties, and the number of losses.

We define the constructor `make-team-record` as follows:

```
(define (make-team-record name wins ties losses)
  (cons name (list wins ties losses)))
```

Write the selectors `team-name`, `team-wins`, `team-ties`, and `team-losses`.

To store all of the teams, we will use the following constructor:

```
(define make-teams-list list)
```

Write the selectors `first-team` and `rest-teams`.

Problem 4 cont.

Write a procedure called `winning-teams` which takes a `teams-list` and returns a list of the records of the teams who have more wins than losses. Be sure to use the constructors and selectors that you helped define on the previous page.

Problem 5 (20 points)

Write a procedure `min-of-f-x-and-g-x` that takes two procedures `f` and `g` and a number `x` as inputs, and returns the minimum of applying `f` to `x` and `g` to `x`. (You may use the Scheme primitive `min`, which returns the minimum of its arguments.)

```
(min-of-f-x-and-g-x square cube -1)
;Value: -1
```

```
(min-of-f-x-and-g-x square cube 2)
;Value: 4
```

Problem 5 cont.

Generalize the `min-of-f-x-and-g-x` procedure so that the procedure you apply to the results of `f` and `g` is a parameter too.

```
(combine-f-x-and-g-x min square cube -1)
;Value: -1
```

Write a procedure `mul-f-x-and-g-x` which uses the `combine-f-x-and-g-x` procedure.

Problem 6 (20 points)

Write a procedure called `add-two-lists` which takes two lists of numbers and returns a list whose elements are the sum of the corresponding elements of the original lists. For example,

```
(add-two-lists '(1 2 3) '(10 20 30))
```

would return the list

```
(11 22 33)
```

You may assume that the lengths of the two lists are equal.

What is the order of growth of your procedure in terms of time and space?

Time: _____

Space: _____

Does your procedure generate a recursive or an iterative process? _____