**Review Questions:**

13, 20, 21, 23, 35, 36, 45

**Problem**

9

**Programming Exercise**

1, 4

RQ:

13.) Why are CAR and CDR so named?

CAR returns the contents immediate from the list address which is the first item.

**C**ontents of the **A**ddress part of **R**egister number

CDR returns the decrement which is the complement of CAR, returning the rest.

**C**ontents of the **D**ecrement part of **R**egister number

These are primitives used to compose other recursive function calls.

20.) In what ways are Common LISP and Scheme opposites?

Scheme utilizes static scoping and is stripped of its bells and whistles whereas Common LISP is fully implemented with dynamic scoping and many more features. This causes Scheme to be a lot smaller and simpler to be used for teaching purposes while Common Lisp is good for commercial use.

21.) What scoping rule is used in Scheme? In Common LISP? In ML? In Haskell? In F#?

Scheme – static

Common LISP – dynamic

ML – static

Haskell – static

Scope – dynamic

23.) What are two ways that ML is fundamentally different from Scheme?

ML is a strongly typed language unlike Scheme is non typed.

35.) What does lazy evaluation mean?

Lazy evaluation means the expressions are only evaluated when their values are needed.

36.) What is a strict programming language?

A strict programming language requires all of its parameters are to be evaluated, allowing the order that they are evaluated does not matter.

45.) What support does Python provide for functional programming?

Python has support for lambdas (anonymous functions) and other functional constructs such as mapping, accumulators, filters, and reducers.

P:

9.) What does this Scheme function do?

(define (y s lis)

(cond

((null? lis) '() )

((equal? s (car lis)) lis)

(else (y s (cdr lis)))))

It goes through the list “lis” and checks if the value stored at the address is equal to s. If it is equal, return the list. If not, pass the rest of the list besides that first item recursively in y. Once lis returns null, return an empty list ‘().

PE:

1. Write a Scheme function that computes the volume of a sphere, given its radius.

(define sphereVolume

(lambda ( r )

( \* ( / 4 3) ( \* pi r r r))))

4.) Write a Scheme function that takes two numeric parameters, A and B, and returns A raised to the B power.

(define (power A B)

(cond

((equal? B 0) 1)

( else ( \* A (power A ( - B 1 ))))))