**ICS 365. Organization of Programming Languages.**

**Programming Assignment 4.**

**Due**: November 13, 2016

**Points**: 40

1. (10 points)

Write a scheme function named up-to-first-number that takes a list as its input and returns a list containing all the elements up to the first numeric element in the input list. You can use the *number?* predicate function to determine whether an element is a number or not.

Sample runs:

(up-to-first-number '(a b c d 1 2 3 )) returns (a b c d)

(up-to-first-number '(d e f 7)) returns (d e f)

(up-to-first-number '(g h i)) returns (g h i)

(up-to-first-number '(1 2 3)) returns NIL

2. (10 points)

Write a scheme function find-loc which takes two parameters, a list lst and an atom atm, and returns the index of the first location where atm occurs in the list.

The location index is 1-relative. If atm does not occur in the list, the function returns n + 1, where n is the length of the list.

3. (20 points)

Write a function named longer-list that takes two list arguments and returns the longer list of the two inputs. If the two lists are equal in length, the function returns #t, and if one of the arguments is not a list, the function should return #f.

**Note: You are not allowed to use the predefined length function; however, you can write your version of** length **or other helper functions that you may want to call from** longer-list**.**

Sample runs:

(longer-list '(1 2 3 4) '(a b c d e)) returns (a b c d e)

(longer-list '(d e f) '(4 5 6)) returns #t (or true)

(longer-list '(g h i) 3) returns #f (or false)