CS345 Programming Languages
Midterm 1 Review

1. Give the types of each of the following ML functions (be careful!):

   a. fun length [] = 0 | length (a::b) = 1 + length b;
   b. fun double [] = [] | double (x::y) = x :: x :: (double y);
   c. fun double2 [] = [] | double2(x::y) = [x] :: (double y);
   d. fun compress [] = [] | compress (x::y) = x @ (compress y);
   e. fun foo (y:real) = fn x => x + y;
   f. val foo2 = foo 5.0;
   g. fun bar f [] = 0 | bar f (a::b) = f(a,(bar f b))
   h. fun odd (a,b) = length(a) + b;

2. Given the above function definitions, give the values of the following expressions.
   a. length [[1,2,3],[4,5],[6,7,8]];
   b. double [1,2,3];
   c. double2 [1,2,3];
   d. compress [[1,2],[3,4]];
   e. compress[[[1,2],[3,4]],[[5,6],[7,8]]];
   f. (foo 3.0) 4.0;
   g. foo2 4.0;
   h. bar (op +) [3,4,5];
   i. bar (op *) [6,7,8];
   j. odd ([1,2,3],10)
   k. bar odd [[1,2,3],[4,5],[6,7,8]]

3. Use map and/or reduce to calculate the following quantities:

   a. The sum of the squares of all elements in an array A
   b. The maximum value stored in an array A
   c. The number of 3's in the list A

4. Write the following ML functions:

   a. **numinrange : int * int * (int list) -> int** numinrange takes as input two integers low and high, and an integer list, and returns the number of elements in the list that are <= low and >= high
   b. **filter : ('a -> bool) -> 'a list -> 'a list** filter takes as input a function f and a list l, an a list, and returns a new list consisting of all the elements e of l such that (f e) = true.
   c. **inboth : ''a list * ''a list -> ''a list** inboth takes as input two lists, and returns a list of all elements that occur in both input lists.

5. Write the following lisp functions:

   a. Write a function **numeven** which returns the number of even elements in a list. Assume the existence of a predicate function even which returns 't for even numbers, and () for odd numbers
   b. Write a function **sublistlen** that takes as an input parameter a list of sublists, and returns the sum of the lengths of the sublists.. So, (sublistlen '((1 2) (3 4 5) (6 7))) would return 7, and (sublistlen '((a b) (c d e f))) would return 6
   c. Write a function **follow** that takes as input paramters a list and two elements e1 and e2, and returns the number of times e2 immediately follows e1 in the list. Thus, (follow 'a 'b '(c b a b b c a b a)) would return 2, and (follow 3 8 (8 3 4 2 3 8 3) would return 1.