CS 603 Exam 2 Extra **Spring 2007** **Solution**

9. Due to ML's type restrictions, some values and functions cannot be written in ML as generally as they can be in Scheme. For example, the list '(5 abc (99 xyz ( ))) is permitted in Scheme, but the corresponding list [5, "abc", [99, "xyz", [ ]]] is not allowed in ML because type inference reports a type error. Using ML, show how to create a simulated representation of this given mixed-type nested list. **[8 points]**

   First define a new datatype that permits values to be ints, strings, or nested lists:

   datatype MYTYPE = INT of int | STRING of string | LIST of MYTYPE list;

   Next create a value of this new type that simulates the mixed-type list given above:

   LIST [INT 5, STRING "abc", LIST [INT 99, STRING "xyz", LIST [ ]]];

10. Write a prefixes function in ML that works as follows:
    prefixes [10, 20, 30, 40] returns [[ ], [10], [10, 20], [10, 20, 30], [10, 20, 30, 40]].

    a. First write prefixes using recursion and pattern matching. Do not use map, foldr, or foldl. **[8 points]**

```
fun prefixes [ ] = [[ ]]
  |  prefixes (h::t) = [ ]::attach h (prefixes t);

fun attach _ [ ] = [ ]
  |  attach x (h::t) = (x::h)::attach x t;
```

    b. Next write prefixes using map, foldr, and/or foldl. Do not use any explicit recursion. **[6 points]**

```
fun prefixes L = foldr (fn (x,y) => [ ]::map (fn z => x::z) y) [[ ]] L;
```