CSci 4651 Fall 2003 Review Questions for Final Exam

Problem 1: type inference. Show the work of type inference algorithm for the following ML functions:

```
fun f(x, y, z) = x(1 + y,z)::nil;

fun f(x) = x::x;

fun f(x) = (x::nil)::nil;

fun f(x, y, z) = x::(y + 3)::z;

fun f(nil) = nil |
    f(true::xs) = 1 + f(xs) |
    f(false::xs) = f(xs);
```

Problem 2. Given the tree datatype

```
datatype 'a tree = LEAF of 'a | NODE of 'a tree * 'a tree;
```

- write a recursive function that counts the number of leaves in the tree
- write a recursive function that creates a "mirror image" of a tree, i.e. a tree in which the left and right subnodes of every node are switched.

Problem 3. Which of the following ML operators are polymorphic, which ones are overloaded, and which ones are neither? Please explain your answer.

1. `::`
2. `^`
3. `+`
4. `=`

Pass-by-value, pass-by-reference. For each of the parameters of the function `f` in the program fragment below figure out if it has been passed by value or by reference, based on the fact that the final value of `n` in `main` is 3. Explain your reasoning.

```
main() {
   int n = 0;
   f(n,n);
   print(n); // prints 3
}

void f(int x, int y) {
   x++;
```

```
   if (x != y) {
      y = 3;
   }
}
```

Problem 4. Function implementation. This is a more convoluted version of Problem 6 in Problem set 6.

As before, assume the static scope rules and call-by-value parameter passing. `int -> int` is the type of functions from an integer to another integer.

```
int x = 0;
int y = 2;

int f(int y) {
    if (y <= 1) return x;
    else {
        int y = 1;
        x = x  +  y;
        return f(y - 2);
    }
}

void g (int -> int h, int x) {
    print(h(x));
}

void main () {
    int x = 2;
    g(f,3);
}
```

Question 1. Note that `f` is recursive. Draw the program stack and function closures when all activation records for `f` needed for this program are pushed on the stack.

Question 2. What is going to be printed in the program? What are the final values of global x and y? Use the stack picture to explain your answer.