CS 257: Non-Imperative Programming: Scheme!
Homework 2 (Spring '06)

A *binary number* is a representation of a number in base 2. It consists of a sequence of binary digits (or *bits*). Each bit b_i either has the value 0 or the value 1. The decimal value of a binary number $b_n b_{n-1} \ldots b_0$ can be computed as follows:

$$b_n 2^n + b_{n-1} 2^{n-1} + \ldots + b_1 2^1 + b_0 2^0$$

where $2^0 = 1$. Binary numbers can be represented in Scheme as lists of boolean values representing bits. The boolean value #f can be used to represent 0 and the boolean value #t can be used to represent 1. The bits appear in the list in *reverse order*. For example, the binary number 10010 (with decimal value 18) can be represented by the list (#f #t #f #f #t). The binary number 0 can be represented by the empty-list and the binary number 1 by the singleton list (#t).

1. Give a definition for a function, *nzero?*, which returns #t if its argument is the binary representation of the natural number zero and #f otherwise.

2. Give a definition for a function, *nadd1*, which given a binary number, returns a binary representation of a natural number which is arithmetically greater by one.

3. Give a definition for a function, *nsub1*, which given a binary number, returns a binary representation of a natural number which is arithmetically lesser by one.

4. Using the definition of *n** given in class, give the binary representation for the natural number which is the product of the natural numbers with binary representations (#f #t #t #t #f #t #f #t) and (#t #t #t #f #t #t).

5. Give a definition for a function *n=*, which returns #t if its natural number arguments are arithmetically equal and #f otherwise.

6. Do the following exercises from Springer and Friedman: 4.1, 4.4, 4.6, 4.10, 4.11