COMP 210, Spring 2001, Homework 4
Due Friday, February 14, 2001 at the start of class

Before you start the homework, you should remind yourself of our General Advice, Advice on Homeworks, and Grading Guidelines. All are available from the class web site (http://www.owlnet.rice.edu/~comp210).

For this assignment, you should follow all the steps of the design methodology and include the results of each step as comments or code in the final materials that you submit. (For example, write your template as a comment—at the appropriate point in the development sequence—and copy it before you fill it in to form your program.)

1. (2 pts) Consider the domain of natural numbers, as defined in Lecture (notes are on-line). Write a program **`multiply`** that takes two natural numbers and returns their product. Your program may not use the built-in multiply function; instead, you should use addition and subtraction to compute the answer.

   Show all the steps in the design methodology. Hand evaluate two cases. Use DrScheme to evaluate them, as well as other test cases.

2. (3 pts) From Lecture 8, recall the definition for **`list-of-sym-and-num`** from Lecture 8 (notes are on-line) and the representation of a recipe as a **`list-of-sym-and-num`**. Develop a program called **`substitute`** that takes three arguments–-a **`list-of-sym-and-num`** that represents a recipe, a symbol **`old`**, and a symbol **`new`**. The program **`substitute`** should create a new recipe in which all occurrences of **`old`** are replaced with occurrences of **`new`.**

   Show all the steps in the design methodology. Hand evaluate two cases. Use DrScheme to evaluate them, as well as other test cases.

3. (5 pts) We can define a **`list-of-list-or-symbol`** as

```
;; a list-of-list-or-symbol is one of
;;    – empty, or
;;    – (cons f r)
;;    where f and r are both list-of-list-or-symbol
;;    – (cons f r)
;;    where f is a symbol and r is a list-of-list-or-symbol

;; Example data
(define lls1
  (cons 'fee
    (cons (cons 'fie (cons  'foe  empty))
          (cons 'fum  empty))))
```

   a) Write a program **`symbol-count`** that takes a list-of-list-or-symbol and returns the number of symbols occurring in the input list. For the example data given earlier, **`symbol-count`** would produce 4.

b) Write a program **`list-count`** that takes a **`list-of-list-or-symbol`** and returns the number of occurrences of **`empty`** in the input list. For the example data given earlier, the program would produce 2. (Every list contains at least one occurrence of **`empty.`**

c) Write a program **`flatten`** that consumes a **`list-of-list-or-symbol`** and produces a new list-of-symbol that has all the symbols from the list-of-list-or-symbol, in their order of appearance. For the example data given earlier, the program would produce

```
(cons 'fee
  (cons 'fie
    (cons 'foe
      (cons 'fum  empty))))
```

4. (Extra Credit: 3pts) We can use lists of symbols to represent finite sets of symbols, provided that we prohibit such lists from containing duplicates. Write a function **`combos`** that takes a list of symbols **`ls`** representing a set and a number **`k`** less than or equal to the length of **`ls`** and returns a list representing the set of all combinations of the set **`ls`** of size **`k.`**