CS109A Problem Set #7
Due 4:30 PM, Friday, 3/10/95

1) (10 points) If the tail of the tail of list L is the empty list, what is the length of L? (Exercise 6.2.4, FCS)

2) (15 points) Suppose a list is allowed to have duplicates. Write an ML function `delete(x,L)` that deletes x from list L. Note that `delete` must return the new list as its value, and that list must not have any occurrences of x.

3) (15 points) Use the function `reduce`, either in the form given in the notes or on p. 104ff of EMLP (please tell which), to write another function that computes the logical OR of a list of booleans. *Warning*: You might think that `op orelse` gives you a prefix function that computes the OR, but it is considered a syntax error by SML/NJ.

4) (20 points) We can represent a queue by an array, provided that the queue does not grow too long. In order to make the operations take $O(1)$ time, we must think of the array as *circular*. That is, the array `A[0..n-1]` can be thought of as having `A[1]` follow `A[0]`, `A[2]` follow `A[1]`, and so on, up to `A[n-1]` following `A[n-2]`, but also `A[0]` follows `A[n-1]`. The queue is represented by a pair of integers `front` and `rear` that indicate the positions of the front and rear elements of the queue. An empty queue is represented by having $front$ be the position that follows $rear$ in the circular sense; for example, $front = 23$ and $rear = 22$, or $front = 0$ and $rear = n - 1$. Note that therefore, the queue cannot have n elements, or that condition too would be represented by $front$ immediately following $rear$. Thus the queue is full when it has $n - 1$ elements, not when it has n elements. Write functions in C for the queue operations *enqueue* and *dequeue*, assuming the circular array data structure. Do not forget to check for full and empty queues. (Part of Exercise 6.8.3, FCS)

5) (15 points) Prove the equivalence $\big((S \cup T) - R\big) \equiv \big((S - R) \cup (T - R)\big)$ by Venn diagrams, or equivalently by building a table showing the 8 classes of elements (in or out of each of R, S, and T). Indicate, for each subexpression, the set of regions of the Venn diagram that it represents. (Exercise 7.3.2(b), FCS)

6) (15 points) Assuming $S \subseteq T$, prove $S \cup T \equiv S$ by showing that each side is contained in the other. (Exercise 7.3.2(a), FCS)

7) (10 points) Suppose we wish to store a set of English words in a hash table with $B = 128$ buckets. Would you expect the hash function $h(x) =$ the ASCII code for the last letter of word x to be a good choice (i.e., "good" = it distributes the words into buckets fairly evenly)? Why or why not? (Exercise 7.6.2(b), FCS)