Com Sci 221 — Programming Languages
Final Exam — Winter 1993
Monday, 15 March
2 pages, 6 problems
Open book, open notes, open mind

Each numbered problem is equally valuable.

1. Give the best example you can of a single procedure declaration and a single call to the procedure that yields 3 different results using call by value-result, call by reference, and call by name. Do not use global variables, and use only assignments to variables in the body of the procedure (in particular, no input nor output). The actual parameters in the procedure call must all represent legitimate l-values, and the call must terminate normally, for all three parameter-passing methods. The difference between the three parameter-passing methods should lead to different values for one or two variables in the main program immediately after the call to the procedure. The quality of your example will be judged according to simplicity, brevity, and especially the clarity with which it illustrates the *reasons* for the discrepancies between the parameter-passing methods.

2. For each of the following pairs of types, state whether the two are equivalent ($\cong$), the first can represent the second ($\succ$), the second can represent the first ($\prec$), or the two are incomparable ($\perp$). You may use the symbols in parentheses to express your answer, or use English phrases if you prefer.

(a)	$(A \times B)^*$	$A^* \times B^*$
(b)	$A \to (B \to C)$	$(A \times B) \to C$
(c)	$(A \to B) \to C$	$A \to (B \times C)$
(d)	$(A^*) \times (A^*)$	$(A^*) \times integer$
(e)	$(A \to C) \times (B \to C)$	$(A + B) \to C$
(f)	$(A \to B) \times (A \to C)$	$A \to (B + C)$

3. Describe in detail the execution of the assignment command A[i]:=A[A[i]]. Mention each calculation of an l-value and each calculation and modification of an r-value explicitly, and make sure that they come in a sensible order. Give an example of values for A and i for which A[i]$\neq$A[A[i]] immediately after execution of the command.

4. What types will ML derive for the following defined functions?

```
fun identity x = x;
fun repeat f x = f (f x);
val stutter = repeat identity;
fun firstarg x y = x;
fun mix f g x = f x (g x);
val mix firstarg firstarg;
```

Hint: the last one is hard, the rest are easy.

5. Suppose that you are given an array A of integers, indexed from 0 to n ($n \geq 2$), and you are guaranteed that the values in the array increase from $A[0]$ up to some maximum value at a position $A[t]$, then decrease from $A[t]$ to $A[n]$. That is, there exists a t, $0 < t < n$, such that for all i, $1 \leq i < t$, $A[i] < A[i+1]$, and for all j, $t \leq j \leq n$, $A[j] > A[j+1]$. (Pay close attention to the difference between strict $<$, $>$ and nonstrict $\leq$, $\geq$.) Write a simple while-loop program to find the value of t, by manipulating the values of three variables, $left$, $middle$, and $right$, using the invariant

$$A[left] < A[middle] > A[right]$$

It is essential to use the given invariant—there is a very different program using a different invariant. How is the value of t determined when your loop halts? In what way would this problem be more complicated if we allow $t = 0$ and $t = n$? (One or two sentence answers should suffice.)

6. What stream is the value of **mystery** in the following Scheme definitions?

```
(define (add-streams s1 s2)
        (cons-stream (+ (head s1) (head s2))
                (add-streams (tail s1) (tail s2))))

(define mystery
        (cons-stream 1
                (add-streams mystery mystery))
```