Com Sci 221 — Programming Languages
Final Exam — Winter 1991
Friday, March 22
2 pages, 7 problems
Open book, open notes, open mind

Each numbered problem is equally valuable.

1. The obvious recursive definition of list reversal is

$$\begin{aligned}
&rev[()] = () \\
&rev[(x\ .\ l)] = addend[l; x] \\
&addend[(); x] = (x) \\
&addend[(y\ .\ l); x] = (y\ .\ addend[l; x])
\end{aligned}$$

It is easy to see that the number of steps to reverse a list of length n is about $n^2/2$. The obvious iterative program takes only n steps:

$l := l_0;$
$m := ();$
while $l \neq ()$ **do begin**
    $m := (car[l]\ .\ m);$
    $l := cdr[l]$
end

Use the idea in the iterative program above to give a purely functional recursive definition of *rev* that takes about n steps to reverse a list of length n. You will need at least one auxiliary function in your definition.

2. If you were designing a programming language, would you make subtraction left-associative or right-associative? Briefly explain why.

3. Consider the following program fragment that computes $\lfloor \log_2 x_0 \rfloor$:

$x := 1;$
$y := 0;$
while $2 * x \leq x_0$ **do begin**
    $x := x * 2;$
    $y := y + 1$
end

Give appropriate precondition, postcondition, and loop invariant to show that the program fragment is correct (they are quite simple). Be careful to include *all* essential information in the invariant. *Use your invariant* to explain why the loop condition must be $2 * x \leq x_0$ rather than $x < x_0$.

4. 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)	$A \to (B \to C)$	$(A \times B) \to C$
(b)	$(A \to B) \to C$	$A \to (B \times C)$
(c)	$(A^*)^*$	$(A + \{0\})^*$
(d)	$\{0,1\} \to A$	$A \times A$
(e)	$(A + B)^*$	$A^* + B^*$
(f)	$(A \times B)^*$	$A^* \times B^*$
(g)	$A \times (B + C)$	$(A \times B) + (A \times C)$
(h)	$A + (B \times C)$	$(A + B) \times (A + C)$

5. Give the best example you can of a single procedure declaration and a 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. 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 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.

6. What type will ML deduce for each of the objects defined in the following fragments?

```
fun identity x = x;
fun interpret nil f x = x
| interpret (y :: l) f x = f y (interpret l f x);
fun mystery l = interpret l identity;
```

Describe briefly but precisely the meaning of the **mystery** function. Hint: what is the value of

```
mystery [e,f,g,h] x
```

7. Do problem 8.2 on pp. 339 of the text. (The definition of `append` is on p. 310.)