Com Sci 221 — Programming Languages
Final Exam — Winter 1990
Monday, March 12
2 pages, 7 problems
Open book, open notes, open mind

Each numbered problem is equally valuable.

1. Draw an abstract syntax tree for the following expression, and also convert it into postfix form, assuming that $*$ has precedence over $+$, and both are left-associative:

$$x + 2 * y * (u + v + 1) + 3$$

   Very briefly, how is postfix relevant to compiling machine code?

2. Describe, in English or in RAM code, the execution of the assignment command A[i]:=A[A[i]]. Give an example of values for A and i for which A[i]$\neq$A[A[i]] immediately after execution of the command.

3. 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)	$A + (B \times C)$	$(A + B) \times (A + C)$

4. Show a procedure definition and a call to the defined procedure that will produce different results for call-by-value-result and call-by-reference.

5. For each of the following problems decide whether mark-and-sweep (the "lazy approach" to deallocation on p. 290 of the text) or reference-count (the corresponding "eager approach") is likely to be more satisfactory, assuming that time is a more critical resource than space. Explain each choice briefly. Concentrate on three issues:

   - interactive response time vs. total time,
   - time proportional to the amount of space used vs. time proportional to the amount of space recovered,
   - time spent when storage is allocated vs. time spent when storage is deallocated.

   For the purposes of this problem, there is no point in considering copying collectors, as their behavior is qualitatively the same as mark-and-sweep.

(a) A desk-calculator program is required to evaluate each arithmetic expression typed in by the user interactively, producing the value of one expression before the user types the next.

(b) A simple function must be applied to each integer in a long list, producing a list of results. The result list must be allocated, but little or no deallocation is expected.

(c) A database system is implemented by storing all data in a huge Lisp S-expression called **data**. This S-expression changes very infrequently, but many queries are processed every day in batch mode. Queries are expressed as Lisp functions that are automatically applied to **data**.

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

```
fun compose f g x = f (g x);
fun repeat f = compose f f;
fun repcomp f g = repeat (compose f g);
fun doublecomp f g = repeat (compose f) g;
fun identity x = x;
fun complist nil = identity
| complist (f :: l) = compose f (complist l);
```

7. Recall the binary search of a sorted array A to find a target value t, using variables l and u as the lower and upper bounds of the portion of the array that is still being searched. Either of the following two assertions may be used as the main part of an invariant:

(a) $A[l] \leq t \leq A[u]$

(b) If there is an i such that $A[i] = t$, then there is an i such that $A[i] = t$ and $l \leq i \leq u$

Which one is more helpful when t is *not* found in A, and the next task of the program is to insert it. Very briefly, why?