Com Sci 221 — Programming Languages
Midterm Exam — Winter 1993
Due Monday, 1 March
2 pages of problems, 5 problems
Open book, open notes, open mind

Take this exam at a time and place of your own choosing, but in time to hand it in at the beginning of class on Monday, 1 March. Give yourself a contiguous 3-hour block of time to do the problems, starting when you open the exam paper. You may refer to your text, notes or any other written materials gathered in advance. You may not communicate with anyone except the instructor about the exam or class material during your exam period. Each numbered problem is equally valuable.

1. Draw the abstract syntax tree and postfix forms for the expressions $(a+(b+(c+d)))*e$ and $a*(((b+c)+d)+e)$. How tall will the stack get when evaluating each of these expressions in postfix form? State a general principle that determines the height of stack required to evaluate a given abstract syntax tree when it is translated to postfix.

2. $A \to B$ denotes the set of functions from A to B (*array A of B*), $A \times B$ denotes the set of ordered pairs from A and B (records with two fields, one containing a value in A and the other a value in B), and $A+B$ denotes the marked union of A and B (variant record containing either a member of A or a member of B). The following 8 types form 4 pairs of essentially equivalent types. What are the pairs?

   (a) $A \times A$

   (b) $A \to (B \times C)$

   (c) $A \times (B + C)$

   (d) $(A \to C) \times (B \to C)$

   (e) $(A \to B) \times (A \to C)$

   (f) $(A \times B) + (A \times C)$

   (g) $\{0,1\} \to A$

   (h) $(A + B) \to C$

3. Let A be an array with indexes in the range $[1, n]$, and integer entries in nondecreasing order (that is, $0 \leq i \leq j \leq n$ implies that $A[i] \leq A[j]$). Use the following loop invariant to write a simple program to do binary search for the *first occurrence* of the value v, by manipulating the program variables *lower* and *upper* (the variables i and j are *not* program variables, they are used only to give information about the contents of A):

   - For i in the range $1 \leq i \leq lower$, $A[i] < v$
   - For j in the range $upper \leq j \leq n$, $A[i] \geq v$

   It is *essential* to use the given invariant—I am not interested in another binary search program using a different invariant. Pay close attention to the difference between strict inequalities ($<$, $>$) and nonstrict inequalities ($\leq$, $\geq$).

   If the first occurrence of v in A is at $A[i]$, what are the final values of *lower* and *upper*? If v does not occur in A, what are the final values of *lower* and *upper*?

4. Draw a simple flowchart that cannot be derived directly from a program involving only assignment, conditional, and **while**-loops. What property of the flowchart makes it impossible that it came from such a structured program?

5. 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. 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.