Com Sci 221 — Programming Languages
Assignment #3 — Winter 1994
Due Wednesday, 26 January

1. We saw in class that, although the stack implementation of expression evaluation is designed primarily for left-to-right innermost evaluation, it allows certain types of outermost evaluation (called "selective evaluation" in the text), such as that required to produce the usual behavior of conditionals under the set of rules

$$if\ T\ then\ \alpha\ else\ \beta \implies \alpha$$

$$if\ F\ then\ \alpha\ else\ \beta \implies \beta$$

and the slightly unusual, but extremely useful, behavior of *and* and *or* under the set of rules

$$F\ and\ \beta \implies F$$

$$T\ and\ \beta \implies \beta$$

$$T\ or\ \beta \implies T$$

$$F\ or\ \beta \implies \beta$$

Comment on the problems involved in trying to do outermost evaluation on a stack with sets of rules such as

(a)

$$\alpha\ or\ T \implies T$$

$$\alpha\ or\ F \implies \alpha$$

(b)

$$\alpha\ or\ T \implies T$$

$$T\ or\ \beta \implies T$$

$$F\ or\ F \implies F$$

(c)

$$if\ \alpha\ then\ \beta\ else\ \beta \implies \beta$$

$$if\ T\ then\ \alpha\ else\ \beta \implies \alpha$$

$$if\ F\ then\ \alpha\ else\ \beta \implies \beta$$

This is a deliberately open-ended question. You will get more credit the more insight you can generate about why each set of rules is not only problematic, but *more* problematic than the ones before. Long answers, however, are not recommended. It is possible in each case to get the essential points in a few sentences.

2. Imagine a small company office with only 10 employees, numbered 1–10 (to imitate the impersonal charm of a large corporation). Each employee has a supervisor, but not necessarily within the local office. The number of the ith employee's supervisor is given by the array element *supervisor*$[i]$, where an entry larger than 10 indicates a supervisor outside the local office. Similarly, the number of employees supervised by employee number i is given by the array element *employees*$[i]$. You wish to find an employee whose supervisor is in the same office, and is in charge of only that one employee. The basic structure of your program to find a solution should be

   $i := 1$;
   While $i \leq 10$, but the ith employee's supervisor is not what you are looking for **do**
       $i := i+1$
   end while

   Realize the structure above as well as you can in Pascal and C. You should do better in C, by using the conditional forms of *and* and/or *or* (written "`&&`" and "`||`"). Demonstrate your programs on a set of inputs that demonstrates the interesting qualities of the programs.

3. Look up the precedence and associativity rules for the binary integer and boolean infix operators of Pascal and C, as well as the unary minus. Compare the two. Find examples of textual expressions that denote different abstract syntax trees in the two languages (ignoring any differences between symbols used for the same operation). Discuss the relative advantages and disadvantages of each interpretation.