ELSE RETURN is no better.

Such convolutions are almost never necessary if decisions are made in the right order.

```
FINDNUM: PROCEDURE OPTIONS (MAIN);
   DECLARE (NEWIN, LARGE) DECIMAL FLOAT (4);
   NEWIN = 0;
   LARGE = 0;
   DO WHILE (NEWIN >= 0);
      GET LIST (NEWIN);
      IF NEWIN > LARGE THEN
         LARGE = NEWIN;
   END;
   PUT LIST (LARGE);
END;
```

What we have here is a simple DO-WHILE, done while the number read is not negative, controlling a simple IF-THEN. Of course we have rearranged the order of testing, but the end-of-data marker chosen was a convenient one and does not interfere with the principal work of the routine. True, our version makes one extra test, comparing the marker against LARGE, but that will hardly affect the overall efficiency of the sequence. Readability is certainly improved by avoiding the ELSE GOTO's.

Avoid ELSE GOTO *and* ELSE RETURN.

Most of the IF-ELSE examples we have shown so far have a characteristic in common. Each approximates, as closely as the programmer could manage, a minimum depth decision tree for the problem at hand. If all outcomes have equal probability, such a tree arrives at the appropriate action with the minimum number of tests on the average, so it might seem desirable to lay out programs accordingly. But a program is a one-dimensional construct, which obscures any two-dimensional connectedness it may have. The minimum depth tree is not the best structure for a readable program.

Recall the program for finding the minimum of three numbers which we showed at the beginning of Chapter 2. Let us rewrite that program in PL/I, adhering to the spirit of the original Fortran, but using only IF-ELSE's:

```
IF X >= Y THEN
   IF Y >= Z THEN
      SMALL = Z;
   ELSE
      SMALL = Y;
ELSE
   IF X >= Z THEN
      SMALL = Z;
   ELSE
      SMALL = X;
```

Even though neatly laid out and systematically indented, it is still not easy to grasp. Not all the confusion of the original can be attributed to the welter of GOTO's and statement numbers. What we have here is a "bushy" tree, needlessly complex in