This program is thoroughly, even excessively, commented. Most of the input parameters are printed out for inspection and verification. But there are several errors that warrant discussion. To begin with, what is the value of E when statement 70 is executed? Again we must search through the code, to find that it has never been set.

On some computer systems, storage is initialized to zero at the beginning of a run; in such cases, this bug should come to light when it is observed that the current is zero for all values of C and V. But if storage is left at some random value, which is true on many systems, the current computed will be wrong (although it might look plausible).

Simple oversight is the most common way to botch initialization. It seems likely, however, that this particular error arose because the programmer was not clear in his mind whether voltage was V (a mnemonic) or E (common usage among electrical engineers, to whom this program is directed).

There are other troubles with this program. Since L is not declared REAL, it is an integer by default. What happens when an integer variable is read in and printed out with a floating point (F) format? We do not know for certain, but we can guess that it will not likely be what was wanted. The missing declaration for L also means that statement 70 contains a mixed-mode expression. If the version of Fortran in use accepts mixed mode, the line will be executed, but probably incorrectly, because it will use whatever has been placed in L as if it were an integer.

Most serious is the error in logic. The program prints the values of C and AI for a series of capacitances, while V is 1.0. Then when C exceeds TC, V is increased by 1.0 (to 2.0), and we return to statement 50. But C is never reinitialized to SC, so after printing *one* value of AI with V equal to 2.0, we immediately increment V to 3.0. We then print one more value of AI (again with C beyond TC), and stop.

It is improbable that this is what the program should do. C has not been initialized to SC each time that V is changed. Whenever you have to build a loop out of spare parts because a DO statement is not suitable, take pains to display clearly how the control parameters are initialized, incremented and tested:

```
      ...
C LOOP ON V
      V = 1.0
   50    WRITE(6,60) V
   60    FORMAT('0', 'VOLTAGE=', F5.0)
C LOOP ON C
         C = SC
   70       AI = V/SQRT(R**2 + (6.2832*F*L - 1.0/(6.2832*F*C))**2)
            WRITE(6,71) C, AI
   71       FORMAT('0', 'CAPACITANCE=', E16.5, '  CURRENT=', E16.5)
            C = C + CI
            IF (C .LE. TC) GOTO 70
         V = V + 1.0
         IF (V .LE. 3.0) GOTO 50
      STOP
      END
```

We will have more to say later in this chapter about the questionable wisdom of using a floating point increment like