```
      SUBROUTINE SORT
      COMMON V, M
      REAL V(50)
      IF (M .LT. 2) GO TO 251
      DO 250 J = 1, M
        T = V(J + 1)
        DO 235 K = 1, J
          I = J + 1 - K
          IF (T .GE. V(I)) GO TO 245
          V(I + 1) = V(I)
235       CONTINUE
        I = 0
245     V(I + 1) = T
250   CONTINUE
251 CONTINUE
      RETURN
      END
```

The outer loop of SORT selects the next element to be inserted among the previous ones. Since there is no previous element for the first member of the array, it is somewhat surprising that the outer DO loop runs from 1 to M instead of 2 to M or 1 to M-1.

As might be expected, this is an error: when J reaches M, T is set to the non-existent V(M+1) and that in turn is inserted somewhere in the array.

The error is a classic off-by-one: the loop is done once too often. Fortunately, it is easy to fix this one. The outer loop should be

```
      DO 250 J = 1, M-1
```

Since Fortran does not permit the limit of a loop to be an expression, we must write instead

```
      M1 = M - 1
      DO 250 J = 1, M1
```

Watch out for off-by-one errors.

A common cause of off-by-one errors is an incorrect test, for example using "greater than" when "greater than or equal to" is actually needed. This program is a binary search routine, which looks for a particular element in a table by halving the interval in which the element might lie, until it ultimately either finds it, or deduces that it isn't present.