```
   DIMENSION X(200),Y(200)
   READ 50, N
50 FORMAT(I5)
 2 READ 51, (X(K), Y(K), K = 1, N)
51 FORMAT (2F10.5)
   READ 52,A
52 FORMAT (F10.5)
   IF (X(1)-A)41, 41, 11
41 IF(A-X(N))5, 5, 11
11 PRINT 53,A
53 FORMAT(1H ,F10.5,
  1     26H IS NOT IN RANGE OF TABLE.)
   STOP
 5 LOW = 1
   IHIGH = N
 6 IF (IHIGH-LOW-1)7, 12, 7
12 PRINT 54, XLOW, YLOW, A, XHIGH, YHIGH
54 FORMAT(1H 5F10.5)
   STOP
 7 MID = (LOW + IHIGH)/2
   IF (A-X(MID))9, 9, 10
 9 IHIGH = MID
   GO TO 6
10 LOW = MID
   GO TO 6
   END
```

First we correct statement 12 to refer to the arrays X and Y, with the appropriate subscripts X(LOW), X(IHIGH), and so on. Presumably this error arose from a careless transcription from mathematical notation to Fortran. The program does not check that N is in range, but it does test whether A is inside the table range, which is good.

What happens if we try to search a table containing only one entry? LOW and IHIGH are both set to one, so we immediately go to statement 7, which sets MID to one as well. Now, since A equals X(1) (A has been tested to be sure it is in the table), we branch to statement 9, where IHIGH is set to one (which does not change it!) and we return to statement 6. LOW and IHIGH are still both set to one, so we immediately go to statement 7.... This program is going to run for a long time.

The problem is that not all possible reasons for terminating the search loop were taken into account. We could patch up this bug with special handling when N is 1, but before we do, let us examine another case. Suppose that the table contains several entries, and that the entry at X(1) happens to match A. Then IHIGH and MID will steadily converge toward 1, while LOW remains at 1. When IHIGH gets to either 3 or 4, MID is set to 2; then since A is less than X(2), IHIGH is set to 2. Now IHIGH-LOW is 1, so the IF at statement 6 sends us to statement 12, and we exit. LOW and IHIGH are left pointing at X(1) and X(2) even though there is an exact match at X(1). We leave it to the reader to decide how pervasive this error is.

Patching is no substitute for rewriting: