anything in BOARD(I,J) but legal values, but errors do happen. Therefore, use the fact that the cases of interest are disjoint, and write

```
IF BOARD(I, J) = 1 THEN
   REDS = REDS + 1;
ELSE IF BOARD(I, J) = -1 THEN
   BLACKS = BLACKS + 1;
ELSE IF BOARD(I, J) ¬= 0 THEN
   PUT SKIP LIST ('ILLEGAL PIECE:', I, J, BOARD(I,J));
```

If you ever get an ILLEGAL PIECE message, you have an early warning of some disastrous bug. At the cost of an occasional extra test and a little extra code, the program limits the spread of nonsense should anything damage the board.

Often a program can be made more resistant to errors at *no* additional cost:

```
GET LIST (N) ;
DO WHILE (N ¬= 0) ;
   GET LIST ((NUMBER(I) DO I = 1 TO N)) ;
   TOTAL = NUMBER(1) ;       /* INITIALIZE FOR SUM */
   DO I = 2 TO N ;
      TOTAL = TOTAL + NUMBER(I) ;
   END ;
   PUT LIST ((NUMBER(I) DO I = 1 TO N), TOTAL) ;
   GET LIST (N) ;
END ;
```

Presumably, the end of input is signaled by reading a zero value for N, so the DO-WHILE carefully tests for this case. Indeed it must, for otherwise the body of the loop will compute and print an incorrect TOTAL when N is zero. But the loop body will also behave incorrectly for negative N, and there is no protection against that.

Any arithmetic comparison can in principle yield three different results — less, equal, greater. Often only two outcomes are reasonable, the third being silly or "impossible." An important aspect of defensive programming is to be alert for these "impossible" conditions and to steer them in the safer direction (assuming the error is so impossible that it's not worth a special check and error message). In this case, the program is less vulnerable if we simply change the loop test to

```
DO WHILE (N > 0);
```

As a general rule, terminate a loop early if the impossible arises, so that infinite loops are avoided.

Program defensively.

Floating point arithmetic adds a new spectrum of errors, all based on the fact that the machine can represent numbers only to a finite precision. Here is a simple example, a program which integrates the polynomial x^2+2x+3 between the limits 1 and 10, by a trapezoidal approximation: