comment underlines the fact that human beings are not likely to know the binary representation of 50. Yet we are expected to recognize a binary 47 on the basis of this one hint. One of the first services to be automated in early computer languages was the conversion of decimal to binary. It would be a shame if we were forced to think in binary, after all these years, by misinformed considerations of "efficiency." (Most compilers will convert "47" to binary at compile time, by the way. Those that will not must certainly provide worse inefficiencies to worry about.)

The proper thing to do here is to introduce a parameter, such as MAXLINES, and initialize it to 47 once and for all at the top of the program. The code becomes much more readable and easier to change. And if there happens to be an expensive conversion, it will occur only once.

Let the machine do the dirty work.

Repeated patterns of code catch the eye when scanning listings. Since the computer is a tool for handling repetitious operations, we should be alerted by such patterns to look for oversights — why didn't the programmer let the computer do the repeating? In the middle of a program for manipulating triangles we see the fragment

```
C     COMPUTE LENGTHS OF SIDES
      AB = SQRT((X2 - X1)**2 + (Y2 - Y1)**2)
      AC = SQRT((X3 - X1)**2 + (Y3 - Y1)**2)
      BC = SQRT((X3 - X2)**2 + (Y3 - Y2)**2)
C     COMPUTE AREA
      S = (AB + BC + AC) / 2.0
      AREA = SQRT(S * (S-BC) * (S-AC) * (S-AB)
      ...
C     COMPUTE ANGLES
      ALPHA = ATANF((4.0*AREA) / (AC**2 + AB**2 - BC**2))
      BETA = ATANF((4.0*AREA) / (AB**2 + BC**2 - AC**2))
      GAMMA = ATANF((4.0*AREA) / (AC**2 + BC**2 - AB**2))
```

We can see immediately the advantage of defining two arithmetic statement functions:

```
SIDE(XA, YA, XB, YB) = SQRT((XA-XB)**2 + (YA-YB)**2)
ANGLE(SAREA, SA, SB, SC) = ATAN2(4.0*SAREA, SA**2 + SB**2 - SC**2)
```

so that we can write

```
      AB = SIDE(X1, Y1, X2, Y2)
      AC = SIDE(X1, Y1, X3, Y3)
      BC = SIDE(X2, Y2, X3, Y3)
       ...
      ALPHA = ANGLE(AREA, AC, AB, BC)
      BETA  = ANGLE(AREA, AB, BC, AC)
      GAMMA = ANGLE(AREA, AC, BC, AB)
```

This is not only easier to write but also easier to modify. For instance the Fortran II name ATANF should be changed whenever possible to the Fortran IV standard ATAN. In fact, the form