```
DECLARE (N,R)FIXED BINARY(31);  ON ENDFILE(SYSIN) GO TO EOJ;
  PUT EDIT('PRIME NUMBER RESULTS')(PAGE,X(13),A);
ST1: GET LIST(N);
  IF N¬>1 THEN
    PUT EDIT('ILLEGAL INPUT N=',N,' <= 1')(SKIP,X(10),A,F(5),A);
  ELSE DO; IF N<=3 THEN GO TO APRIME;
    IF N=2*FLOOR(N/2) THEN  NOPRIME: PUT EDIT(N,' IS NOT A PRIME ',
        'NUMBER')(SKIP,F(15),2 A);
    ELSE DO; DO R=3 TO SQRT(N) BY 2; IF N=R*FLOOR(N/R) THEN
        GO TO NOPRIME; END /* OF DO LOOP */;
APRIME: PUT EDIT(N,' IS A PRIME NUMBER')(SKIP,F(15),A); END; END;
  GO TO ST1;
```

The attempt to squeeze the program into only a few lines has made it hard to read, and concealed the convolutions of the code. Try to find the label NOPRIME, and the executable statement on the same line as a declaration. In the construction

```
IF N¬>1 THEN
  PUT EDIT('ILLEGAL INPUT N=',N,' <= 1')(...)
```

why is the test written differently from the printed output? That makes it necessary to understand two things instead of one. Separating the semicolon from its statement in

```
END /* OF DO LOOP */;
```

although harmless enough here, is prone to error. Similarly the statement

```
                                    PUT EDIT(N,' IS NOT A PRIME ',
'NUMBER')(SKIP,F(15),2 A);
```

uses the card up to its right boundary to no purpose. The alphabetic string still has to be split onto the next card, which in turn demands the "2 A" format item. Put it all on one line. Our version of this program is in Chapter 7.

---

*Format a program to help the reader understand it.*

---

The single most important formatting convention that you can follow is to indent your programs properly, so the indentation conveys the structure of the program to the reader at a glance. Indentation must be done carefully, however, lest you confuse rather than enlighten.