```
N1 = INDEX(TEXT,';');     /*INDEX OF ; IN STRING TEXT*/
NAME = SUBSTR(TEXT,1,N1-1);  /*SUBSTRING IN TEXT FROM 1 TO N1 MINUS1*/
N2 = LENGTH(NAME);        /*LENGTH OF STRING NAME*/
```

and

```
S = S + F * EXP (-(I*B/N)**2/8);
     /* S = S + F * E **(-(I * B/N) SQUARED/8 */
```

and

```
ILOOP: DO I = 1 /* BY +1 ASSUMED */ TO 2*N;
```

Don't just echo the code with comments — make every comment count.

Comments should help the reader over the difficult spots in a program. But when a comment becomes too involved, ask whether the code itself is at fault. We have already seen, in Chapter 3,

```
DCL NEWIN DEC FLOAT (4);
    LARGE DEC FLOAT (4) INIT (.0E1);
/* .0 x 10**1 = .0 x 10 = 0.0                    */
```

where the comment "explains" the ill-considered initialization of LARGE. A bigger example is

```
  /* WE NEED A LOOP TO PRINT      */
  /* EACH LINE BECAUSE WE ARE     */
  /* PRINTING COLUMNS AS ROWS.    */

DO J=1 TO 20;

PUT SKIP EDIT(COL(J),(THRUST(I,J)DO I = 1 TO
        10))(R(FORM));

FORM: FORMAT(X(5),A(9),X(4),10(X(4),F(7,1)));

  /* WE CAN OMIT THE 'END'        */
  /* STATEMENT FOR THIS LOOP      */
  /* BECAUSE WE ARE ABOUT TO END  */
  /* THE PROCEDURE.               */
  END CORRECT;
```

The first comment is incorrect, for we can certainly write

```
PUT EDIT ((COL(J), (THRUST(I,J) DO I = 1 TO 10) DO J = 1 TO 20))
      (SKIP, X(5), A(9), X(4), 10 (X(4), F(7,1)));
```

Since the format FORM appears only once in the program, we have moved it inside the PUT as well, en route adding the right parenthesis missing from the original.

The second comment uses four lines in place of the four characters

```
END;
```

Should the code be changed, the END will probably have to be added anyway.