```
      DOUBLE PRECISION FACTOR, X
      NAMELIST /OUT/ I, FACTOR
      FACTOR = 1.0D0
      DO 100 I = 3, 50, 2
         FACTOR = FACTOR * FLOAT(I*(I-1))
         WRITE(6,OUT)
  100 CONTINUE
      STOP
      END
```

Students of complexity theory will recognize that the first version requires computing time proportional to n^2; the second takes time proportional to n. The absolute amount of computer time saved in this specific case is obviously irrelevant, but the gain in intelligibility is significant.

The author of the factorial program, by the way, included some of the machine-generated answers from his program. The value of 3! is given as

```
5.999999999999999
```

Other values, also integers, are printed just as badly. The I/O routines provided by this particular compiler (*not* by the textbook author) are a typical example of false economy (that is, misplaced efficiency), since they do not produce the most meaningful answer for the user. Not only did the routine distort what was almost certainly an exact floating point 6.0 in its haste, but it then decided (presumably) that it was "too inefficient" to round the decimal representation before printing it.

The Euclidean Algorithm computes the greatest common divisor of two integers KA and KB by a series of divisions. Here is part of a program to do it. (KA and KB are positive.)

```
      IF(KA-KB) 5, 5, 4
    4 KR = KA
      KA = KB
      KB = KR
    5 IF(KA) 6, 7, 6
    6 KR = KB - KB/KA*KA
      KB = KA
      KA = KR
      GO TO 5
    7 PRINT 102, KB
```

Mathematicians have grown used to assuming that KA is less than or equal to KB in the algorithm, and so when the program is implemented, the first four lines of code make sure this is the case. But a moment's reflection shows that if KA is greater than KB, the algorithm works anyway, since the first pass through the procedure does the reversal. Removing the explicit interchange shrinks the code by a factor of two, without increasing its complexity. At the same time we can use the MOD function to improve the readability: