tionally, with minor changes to line 110, you can eliminate the ",8". It will be assumed!   —Shawn K. Smith
Bronx, NY

FILE CAMOUFLAGE

Have you ever had a kid sister or brother, or for that matter anyone, snooping around your diskettes? Here is a simple way to stop them from meddling with your files. When you normally save a program you type SAVE"progname",8 and the directory will list the file as a program (PRG) file. BUT if you type SAVE"progname,s",8 the program will be listed in the directory as a sequential (SEQ) file. The only way to regain access to the file is to load it as follows: LOAD"progname,s",8. This way only people who know your program is not really a sequential file will be able to gain access to it. You can also use this idea with "progname,u" replacing "prognam,s". The disk directory will list the program as a user (USR) file.
—Wesley Vriend
Houston, B.C.

IN-CODE LOAD

The LOAD instruction in the Commodore 64 has a strange anomaly. When used inside a program, program flow does not return to the next statement, but back to the beginning of the program! LOADs inside a program are most often used to load in some accessory ML program. For example, consider the following bit of code:

```
10 LOAD "ML01",8,1
20 PRINT "READY?":REM CONTINUE...
```

If you run this code, line 20 will never be executed. Most people avoid this with:

```
10 IF A=0 THEN A=1:LOAD "ML01",8,1
20 PRINT "READY?":REM CONTINUE...
```

The following bit of code shows a better way, plus it allows for complete freedom in relocating the load.

```
10 OPEN 1,8,0,"ML01,P,R"
20 POKE 185,X:REM X=1 USE LOAD ADDRESS S
TORED WITH FILE, X=0 RELOCATE LOAD
30 POKE 780,0
40 POKE 781,LA:POKE 782,HA:REM LA, HA LO
W AND HIGH BYTE OF NEW LOAD ADDRESS
50 SYS 65493:CLOSE 1
```

It seems like more work, but it is much easier to debug than the other way, especially if the loads are scattered throughout the program.

Here is a small program which allows memory dumps to disk from inside a BASIC program:

```
10 OPEN 1,8,1,"ML01,P,W"
20 POKE 193,SL:POKE 194,SH:REM SL,SH LOW
AND HIGH BYTE OF START ADDRESS
30 POKE 174,EL:POKE 175,EH:REM EL,EH LOW
```

```
AND HIGH BYTE OF END ADDRESS
40 SYS 62957:CLOSE 1
```

—Don Lewis
Folsom, PA

STRUCTURED BASIC?

Teaching students to write structured BASIC programs would be much easier if the computer would allow multiple spaces for indentations after a line number.

This IS possible on the Commodore 64. The line number can be followed by [COMMODORE J]. The remainder of the line can have any number of spaces left before the first keyword for indentation. The graphic symbol which appears the first time does not show up when you go to list the program. But the extra spaces disappear if the line is edited after LISTing (unless COMMODORE J is inserted again). Somehow, this simple procedure does not sink in on many students. However, an even simpler technique does: following a line number with a colon (:) has exactly the same effect as the COMMODORE J. The colon does show up when the program is LISTed, but editing the line does not eliminate the extra spaces.
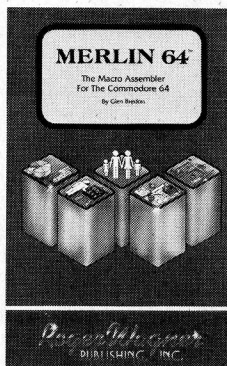—Jack Ryan
El Dorado, AR

Nov. 1985