of RAM beginning at 14336, and line 50 sets the top-of-memory pointer below this block, to prevent BASIC from overwriting it. In this way, for the sake of protecting two Kbytes, we are cutting user memory by two-thirds. This presents no problem while running the character generator program, but will be a potential source of difficulty if we load the new character set to that address for use by an applications program that needs more than 12 Kbytes of user memory. Unfortunately, the mechanics of the operating system prevent our locating the character set any higher in memory — if this was possible, we could put it up in the highest two Kbytes of user memory or in the special programs area at 49152 onwards. The solution seems to be to put the character set as low as possible, and move BASIC above it! This can be done by adjusting the contents of the TXTTAB pointers, but it cannot be done from within a BASIC program, and it must be done before the character generator program is loaded into memory.

The sequence of actions, then, is:

1) LOAD and RUN Program 1. This prints the necessary relocation commands onto the screen, so that you can execute them in direct mode, by pressing Return.

2) LOAD the character generator program and make the following changes:

```
61100 CGEN=53248:NCGEN=2048
61500 POKE PO,(PEEK(PO)AND240)OR2
```

and delete line 50.

3) SAVE this new version.

4) LOAD and RUN the character generator exactly as before.

5) When you finish with the program, LOAD and RUN Program 2. Like Program 1, this prints commands on the screen for you to execute.

6) The TXTTAB and VARTAB pointers were set by Program 2, so SAVE "filename" saves the entire two Kbyte character set area between 2048 and 4097. In future, to run the character generator program you must repeat this sequence, except for step 2.

When you want to retrieve the character set, you must LOAD and RUN Program 1, to move BASIC up in memory, and then LOAD the character set thus:

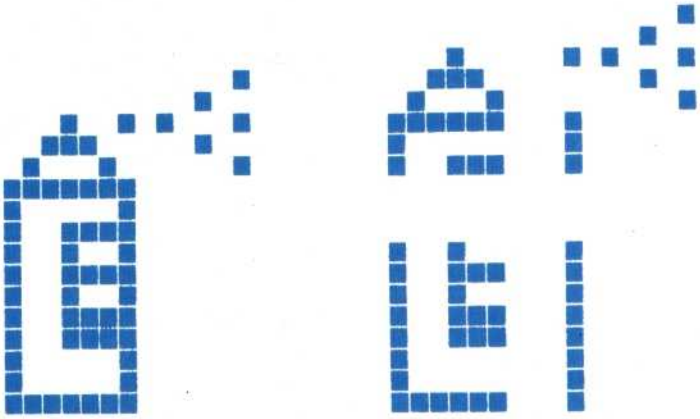
```
LOAD"filename",DN,1
```

where DN (device number) is equal to one for cassette use and eight for the disk drive. The ',1' on the end of the command is known as the secondary address, and is the Commodore way of sending command parameters to peripheral devices. Here, it means that the file is to be loaded to the place in memory from which it was saved, rather than being directed by the TXTTAB pointer into the current BASIC program area. This is possible because when a file is SAVEd the operating system SAVEs the RAM start address as the first data item in the file. When you use the unadorned LOAD command, the start address in the file is ignored in favour of the address pointed to by TXTTAB.

value of X =	bits 3,2,1,0	location pointed to
0	0000	0
2	0010	2048
4	0100	4096
6	0110	6144
8	1000	8192
10	1010	10240
12	1110	12288
14	1110	14336

Value Chart

The command POKE 53272,(PEEK(53272)AND 240)OR X forces the screen display chip to look at the area of RAM containing the redefined character set. The table show the values of X that set the start address of the RAM block.



Once you've re-located BASIC and loaded the new character set, you must make the operating system point at the new character set; this is explained in the table, and is demonstrated in the new version of line 61500 given in this article.

```
199 REM ***************************
200 REM*          PROGRAM 1          *
201 REM*       RUN THIS PROGRAM      *
202 REM*     THEN HIT RETURN TWICE   *
203 REM*    THIS MOVES BASIC TO 4096 *
204 REM ***************************
300 PRINT CHR$(147):PRINT:PRINT
400 PRINT"POKE43,0:POKE44,16:POKE45,3:POKE46,16"
500 PRINT"POKE4096,0:POKE4097,0:POKE4098,
0:CLR:NEW"
600 PRINT CHR$(19)
```

```
199 REM ***************************
200 REM*          PROGRAM 2          *
201 REM*       RUN THIS PROGRAM      *
202 REM*     THEN HIT RETURN TWICE   *
203 REM*     THIS RESETS BASIC PTRS  *
204 REM ***************************
300 PRINT CHR$(147):PRINT:PRINT
400 PRINT"POKE43,0:POKE44,8:POKE45,1:POKE46,16"
500 PRINT"POKE4096,0:POKE4097,0:POKE4098,0:CLR"
600 PRINT CHR$(19)
```