that you know will fit in no more than 3 pages of memory for each group. At the end of each group, insert a go-to instruction using the actual address of the next 3-page segment of memory to follow that group. If your first group starts @ 0200, for example, 04FF would be the last byte in that 3-page area (whether it is used or not). Therefore enter a Go-To 0500 in the last line of that group. You will be assembling the first group at 0200 and the second group at 0500, etc., until all groups are assembled. Loading them in together then will form your long program. With care you can avoid gaps in between groups.

But, you say, how does the first group know where to jump to if that jump goes to the middle of a different group? Easy, but pay attention.

1) Load Chip-8 Assembler-3. Using the ROM System Monitor, change the byte at 0034 to 2A.

Everytime you press Key 2, instead of going to pass 2, the assembler will now expect you to enter a new start address -- you have just disabled pass 2.

Does your program contain more than 14 sections of source code? If not, then continue. Otherwise perform