TIPS 'N TECHNIQUES

Immediate Macros

by Harry White
7495 West 81st Ave
Arvada, CO 80003

Now, let's see. **POKE -16300,0** switches to page one of Apple **high** resolution graphics. Or is it **POKE -16299,0**? Or is that for **low** resolution mode? Now where is that piece of paper I wrote those POKEs on the other day? Where are they in the manual---? "#$%&'!!

I was beginning to think I would have been better off paying the $40.00 for that fancy spelling drill I saw at the computer shop the other day. In the Applesoft program I was trying to write for my kids, the "reward" routine needed some switching back and forth between page one and page two of the graphics screens. The above monologue took place while I was trying to check the thing out in **immediate** mode. It was giving me a little trouble. Certainly, a good graphics utility would have solved my problem — but that would have cost twice as much as the spelling drill for my apple.

There is a way out of such problems for those of you who are as cheap or as poor as I am, or for those who like to solve such things for yourselves. It takes little knowledge of machine language, really, and relies only on the Applesoft commands found in the standard Apple manuals. You need only program complicated maneuvers once, and then you can **use them in immediate mode** whenever you wish.

The concept depends almost entirely upon the use of **RUN** and **END**.

RUN AND END

The **RUN** command in Applesoft, and in many other BASICs, can be given as **RUN<linenumber>**. The program can proceed from that line until it reaches the last program line or until it reaches an **END**. Although Applesoft does not require the END statement at the end of your program, it is often needed to keep the main loops from crashing into the subroutines so often placed at the "bottom" of your works.

FOUR-IN-ONE

These two commands, together, give you the capability to **have a number of separate small programs within a single larger one**. The Apple, DOS, and a tape storage system will all recognize the conglomeration as one whole piece. For example:

```
1 TEXT:HOME:LIST: END: REM** DISPLAY
  MENU
2 POKE -16300,0: POKE -16302,0:
  POKE -16297,0: POKE -16304,0:
  POKE 230,32: END: REM** SWITCH
  TO PAGE ONE HGR
3 POKE -16299,0: POKE -16302,0:
  POKE -16297,0: POKE -16304,0:
  POKE 230,64: END: REM** SWITCH
  TO PAGE TWO HGR
4 FOR N=1 TO 100: A=PEEK(-16336):NEXT:
  END: REM** MAKE BUZZ
```

These are 4 short programs in Applesoft which look like one program to the Apple system. Yet, **they can be RUN individually from the immediate mode** and will put you back in control when each is done. **Line 1** simply LISTs the routines available when the command, **RUN 1**, is typed. From that listing you can choose what to RUN next.

RUN 2 will switch to full page one of high resolution graphics from whatever display mode you are in at the time without clearing any of the screens.

RUN 3 does the same for page two and **RUN 4** gives you a rather sickly little buzz.

By using GOSUB you can expand these ideas to include quite complex routines which are too long to place on one line:

```
5 GOSUB 63000: END: REM** MOVE HI-RES
PAGE ONE TO PAGE TWO
63000 <Start your favorite move routine here.>
63XXX RETURN
```

LOW LINES ARE BEST

Two reasons suggest that you keep your macro line numbers low. **First**, it keeps your keystrokes to a minimum. **Second**, if you are building a program and start the main part of your work with line 100 or higher, the macro part can easily be deleted when you are finished. Macro subroutines, if required, can be given relatively high numbers so that the finished program will fit in between them and the low numbered calling lines. You can then delete these macro subroutines also. Of course, the macros can be included in your main program, although you may need a renumbering utility to do so.

Keep the macro series of routines on a disk. Load them when you begin to program. As you add to them, you will acquire a useful library of macros which fit the kind and style of work you do. You may also save yourself a lot of debugging when using long and complicated POKEs, PEEKs, and subroutines.

We have become used to brilliant, complex and expensive utility programs, sometimes tending to forget such simple alternatives as these immediate macros. You may, however, find yourself swearing less at these than at the "biggies."