

normal way, and force a jump back to \*EXECUT\* in the main program, for example: 95 IF NREC=0 THEN 80. This approach was tried, and worked. But the coding started to look untidy. In accordance with our principle of avoiding GOTOs, we decided to keep things simple and just return to the main menu if a record is not found by \*FNDREC\*.

A small addition to the line 10490 in \*MODNAM\* should be noted. Numeric variable S should also be reset (LET S=0). Failure to do so can, under certain unusual circumstances, cause \*MODNAM\* to malfunction.

The other routine implemented in this final version of the program is \*MODREC\*. This routine first locates the record to be modified by calling \*FNDREC\* (line 14120). This line calls line 13030, not 13000, in order to suppress \*FNDREC\*'s clear screen statement. If the record cannot be located, the program will return to the main menu in the usual way (in line 14130). If the record is located, the target record is left displayed on the screen and users are instructed to:

**MODIFY NAME?  
PRESS RETURN TO ENTER NEW NAME  
OR SPACE BAR FOR NEXT FIELD**

The routine that finds out which of the two options is required can be found in lines 14190 to 14280.

Lines 14190 to 14220 constitute a simple loop that terminates only if either the space bar or RETURN is pressed. If AS is NOT CHR\$(13) (the ASCII value for a carriage return) AND NOT a space (you could also use CHR\$(32) instead of " ") I will be reset and the loop will repeat. If the key pressed was RETURN (i.e. the name field is to be changed) the next few lines will fill the NAMFLDS(CURR) with the new name, set RMOD, reset SRTD, call \*MODNAM\* and fill MODFLDS(CURR) with the standardised name created by \*MODNAM\* and located in MODFLDS(SIZE).

The rest of \*MODNAM\* works in exactly the same way. Note, however, that modifying the other fields does set RMOD but does not reset SRTD (see line 14490, for example). The reason for this is that only changing the name field implies that the data file may be out of order, since the file is ordered by name. Changing any other field merely indicates that a record has been changed (RMOD = 1) and that the file must be saved when the program is terminated.

The other routine implemented is \*DELREC\* — to delete a record. This is very straightforward. First it clears the screen (line 15020) and displays a message explaining what's going on. It then calls \*FINDREC\* to locate the record to be deleted. A choice is then offered: to press RETURN to delete the record or the SPACE BAR to return to the main menu. A warning message is also displayed (line 15160). An even better approach might be to respond with an ARE YOU SURE? message if RETURN is pressed and then only delete the record if the Y key is pressed (i.e. IF INKEYS = "Y" THEN ...).

\*DELREC\* does not reset the SRTD flag. Since the file is already in alphabetical order by name,

deleting a complete record will not upset this order. It does, however, mean that the file has been modified and so RMOD is reset in line 15340 and SIZE is reduced by one in line 13550 to take account of the fact that the file now has one fewer valid records. All the records are moved 'down one' in lines 15260 to 15320.

You may also have noticed that \*FNDREC\* includes a conditional call to a subroutine called \*LSTCUR\* to print out the CURRENT record located by \*FNDREC\*. If you don't have a printer, simply replace line 13540 with a REM for future implementation and omit lines 13600 to 13690.

This completes the address book program. We have carried out all the major options presented in the main menu: finding a record, adding a record, changing a record, deleting a record, and exiting from the program. The purpose of the computerised address book has been to illustrate how a programmer should set about specifying, designing and implementing a program. An essential modification by anyone who intends the program as a piece of application software will be to check for — and trap — the problem that would arise if SIZE were ever to equal 51. This would happen as soon as there were 50 records in the file.

In the next instalment of the Basic Programming course we will discuss programming style and cover a few of the more advanced aspects of the BASIC language.

## Basic Flavours



This command is not available on the Commodore 64, Vic-20, BBC Micro, or Dragon 32.

On the BBC Micro with a parallel printer insert the following lines:

```
13605 VDU 2
13680 VDU 3
```

These enable and disable the printer in turn. Substitute PRINT for LPRINT in lines 13510 to 13670. For more information see the user manual.

On the Commodores insert these lines:

```
13605 OPEN 4,4:CMD 4
13680 PRINT #4:CLOSE 4
```

These enable and disable the printer in turn. Substitute PRINT for LPRINT in lines 13510 to 13670.

On the Dragon 32 insert these lines:

```
13605 OPEN "0", -2
13680 CLOSE -2
```

These enable and disable the printer in turn. Substitute PRINT -2, (the comma here is part of the command) for LPRINT in lines 13510 to 13670.



The address book program for the Spectrum will be published in full in the next instalment of the Basic Programming course.