Chapter 3

Screen Editing

CBM computers display characters on the video screen as they are input via the keyboard. Anything displayed on the screen may be edited or modified in immediate mode.

Screen editing is one of the most significant capabilities of your CBM computer. **You can change your input simply and efficiently using a built-in screen editor**, described in this chapter. Try all the examples provided in this chapter. If you do not understand these editing concepts read Chapter 4, "Programming the CBM Computer," and then reread this chapter. Chapter 4 will give you a fundamental understanding of the BASIC language.

The screen editor allows the cursor to be moved around the screen in four directions. Characters can be inserted or deleted anywhere on the screen. Cursor keys were described in Chapter 1 under "CBM Key Groups": The CLEAR SCREEN/HOME key moves the cursor to the beginning of the top screen line and/or blanks the screen. The CURSOR UP/DOWN, and CURSOR LEFT/RIGHT keys skip the cursor over text. The INSERT/DELETE key adjusts space on the screen to insert or delete single characters.

EDITING TEXT ON THE CURRENT DISPLAY LINE

Often while entering text you may notice a mistake on the line currently being entered. You can correct the mistake immediately. Backspace the cursor to the mistake using either the CURSOR LEFT key or the DELETE key.

We will use the following statement to illustrate editing:

```
MY PET BYTES
```

BITES is misspelled; we want to change the Y to an I.

You can use the CURSOR LEFT key to backspace the cursor back to the Y without altering the text, or you can use the DELETE key to backspace and erase text up to the Y. The choice depends upon whether you want text to the right of the Y to remain. If the error is many characters back, you are better off using the CURSOR LEFT key, which backspaces the cursor, so that you will not have to retype the remainder of the line, possibly introducing new errors.

Backspacing with the CURSOR LEFT Key

Enter the following text, leaving the cursor at the end of the text. Do not press the RETURN key:

```
MY PET BYTES
```

The cursor must be moved back to the Y, which must be changed to an I. BYTES then becomes BITES. To move the cursor left, press the CURSOR LEFT key and the SHIFT key simultaneously one time for each screen position to be moved. Stop pressing the cursor keys when the cursor is positioned at the Y character.

On the CBM 8000 models the cursor will move automatically if the cursor key is held down. You need not repeatedly press the cursor keys. Just hold the key down until the cursor is positioned on the Y character.

```
MY PET BYTES  <—Press CUROSR LEFT
MY PET BYTES  <—Press CURSOR LEFT
MY PET BYTES  <—Press CURSOR LEFT
MY PET BYTES  <—Press CURSOR LEFT
MY PET BYTES
```

To change the Y character to an I, press the I key on the keyboard. Replacing the Y with an I, the cursor moves one position to the right. This is called "typeover," because the cursor types over the existing character with a new character. Typeover changes the screen display to:

```
MY PET BITES  <—Y character typed over with I key.
```

You can move the cursor past the end of the text with multiple pressings of the CURSOR RIGHT key (unshifted) to continue entering more text on the line. Or you can simply press the RETURN key to complete the line. Experiment with both methods.

As each screen line is edited, the change(s) become permanent when you press the RETURN key. Do not move the cursor off the line by pressing either the HOME, CLEAR SCREEN, or CURSOR UP/DOWN key. Although the screen display shows the modification, the change will not be registered in the computer's memory until the RETURN key is pressed.

Backspacing with the DELETE Key

Type in the example text again, leaving the cursor at the end of the text. Do not press the RETURN key.

```
MY PET BYTES■
```

Press the DELETE key one time for each position to be moved. Stop the cursor when the Y character has been erased.

```
MY PET BYTES■ ←— Press DELETE
MY PET BYTE■ ←—— Press DELETE
MY PET BYT■ ←——— Press DELETE
MY PET BY■ ←———— Press DELETE
MY PET B■
```

The cursor should be positioned over the Y character. Press the I key to enter the I character.

To complete the statement you must retype the remainder of the text that was deleted. When completed, press the RETURN key to make the change permanent in computer memory.

```
MY PET B■
MY PET BI■ ←———— Press I key
MY PET BIT■ ←——— Press T key
MY PET BITE■ ←——— Press E key
MY PET BITES■ ←——— Press S key
```

Shifting and Deleting Text with the DELETE Key

The DELETE key can also be used to shift text one position to the left, while simultaneously deleting the character to the cursor's left.

The example statement has been altered to include an extra character in the text, as shown below:

```
MY PET BITTES■
```

Type in this statement, leaving the cursor at the end of the line. Our task is to delete the extra T in BI(T)TES. We will arbitrarily choose to delete the first T. Instead of deleting the text back to the first T, move the cursor over the text with the CURSOR LEFT key. Move the cursor to the second T, because DELETE deletes the character to the immediate left of the cursor, not the character under the cursor.

```
MY PET BITTES
```

With the cursor on the second T, press the DELETE key. The first T is deleted and the text to its right is shifted one position to the left, filling its space.

```
MY PET BITTES ←—— Position of cursor before DELETE key pressed
MY PET BITES ←—— DELETE key pressed
```

Either press the CURSOR RIGHT key several times to move beyond the text before pressing the RETURN key, or simply press the RETURN key to exit the text.

Shifting Text with the INSERT Key

The INSERT key opens a space in the text at the current cursor position, moving all the text beyond the cursor position one space to the right. An additional character may be inserted into this new space.

To demonstrate the INSERT key, we will omit one character from our example text. Type in the statement below, leaving the cursor at the end of the statement:

```
MY PET BTES■
```

The missing I may be added to the text in several different ways.

You can use the DELETE key to delete the text back to the error, then enter the omitted character and the remainder of the line, as previously described. But if you delete more than a couple of characters, you may make mistakes retyping the line. You do not have to retype if you use the INSERT key.

Before using the INSERT key, move the cursor to the character position where you want the insertion to begin. Place the cursor on the character that ultimately becomes the first character *beyond* the insertion.

```
MY PET BTES ←— Position of cursor before INSERT key pressed
        └——— INSERT I character here
```

To insert an I between the B and the T, backspace the cursor by pressing the CURSOR LEFT key until the cursor is on top of the T character:

```
MY PET BTES■ ←— Press CURSOR LEFT
MY PET BTES ←— Press CURSOR LEFT
MY PET BTES ←— Press CURSOR LEFT
MY PET BTES
```

Press the INSERT key once to insert one space between the B and the T. The TES text moves one space to the right while the cursor remains stationary.

```
MY PET B■TES ←— INSERT key pressed
```

Type in the I character:

```
MY PET BITES ←— I character pressed
```

If you want to add text at the end of the line, press the CURSOR RIGHT key to move the cursor beyond the S, add new text, then press RETURN. Or just press the RETURN key to drop the cursor to a lower line if no new text is to be added.

This time we will insert an entire word into a line of text. Type in the following new text, leaving the cursor at the end of the statement:

```
NOW IS THE TIME■
```

Suppose we want to insert the word NOT, to change the meaning of the statement.

```
NOW IS THE TIME■
    ‾‾‾
    NOT
```

Press CURSOR LEFT repeatedly until the cursor backspaces to the T of THE.

```
NOW IS THE TIME■
NOW IS THE TIME
NOW IS THE TIME
NOW IS THE TIME
NOW IS THE TIME
NOW IS THE TIME
NOW IS THE TIME
NOW IS THE TIME
NOW IS THE TIME
```

Press the INSERT key four times to make room for the word NOT, plus a space.

```
NOW IS THE TIME ←——— Press INSERT key
NOW IS ■THE TIME ←——— Press INSERT key
NOW IS ■ THE TIME ←——— Press INSERT key
NOW IS ■  THE TIME ←——— Press INSERT key
NOW IS ■   THE TIME
```

Type in the word NOT and a space:

```
NOW IS ■   THE TIME
NOW IS N■  THE TIME
NOW IS NO■ THE TIME
NOW IS NOT■THE TIME
NOW IS NOT THE TIME
```

Press the RETURN key to exit the cursor from the text.

There are a couple of rules that must be observed when using the INSERT key. Always move the cursor to the position where you want the insertion to *begin*. The character under the cursor will be moved to the right of the cursor, becoming the first character beyond the insertion. When entering in additional characters, press as many character keys, including spaces, as you did INSERT keys.

EDITING TEXT WITHIN QUOTATION MARKS

Editing text enclosed within quotation marks calls for different procedures, because quotation marks signal the beginning or the end of a text string.

Recall from Chapter 2 that any text entered after an odd number of quotation marks becomes a text string; this includes cursor keys. If you press a cursor control key to edit the string, the cursor will not move. Instead, a symbolic representation of the cursor key will be displayed as part of the string. **Before the string can be edited, it must be completed with a second set of quotation marks, or by pressing the RETURN key. Once out of the string, the cursor keys function normally. The CBM 8000 ESC key cancels the effect of an odd number of quotation marks.**

The example text we will use to demonstrate editing within a string is:

```
PRINT "MY PET BITES"
```

This is a PRINT statement which will display the character string MY PET BITES on the next line when the RETURN key is pressed.

When entering this statement we discover the I of BITES was accidentally entered as a Y.

```
PRINT "MY PET BYTES"■
```

To edit the Y character without altering the rest of the text, your first inclination is to press the CURSOR LEFT key three times and type over the Y with an I character. This will not work. The three CURSOR LEFTs will be incorporated into the text string instead of moving the cursor left. The incorporated CURSOR LEFTs, shown by their symbolic representations in the string, cause cursor movement when the PRINT statement is executed.

```
PRINT "MY PET BYTE■■■■ ←— CURSOR LEFT pressed three times
```

Table 3-1 shows the string representations of all the cursor keys. To avoid "programming" the cursor keys in the string, type a second set of quotation marks before you start editing.

Returning to the original statement, to avoid entering the CURSOR LEFT keys as text, enter the remainder of the string, and the second set of quotation marks which end the string:

```
PRINT "MY PET BYTE■
PRINT "MY PET BYTES■
PRINT "MY PET BYTES"■ ←— Cursor positioned after second
                         set of quotation marks
```

Table 3-1. String Representation of Cursor Keys

Function		Key	String Symbol
DELETE		INST DEL	(Reverse shifted T)
INSERT	Shifted	INST DEL	Not programmed
Home Cursor		CLR SCREEN HOME	(Reverse S)
Clear Screen	Shifted	CLR SCREEN HOME	(Reverse Shifted S)
Cursor Down		CURSOR	(Reverse Q)
Cursor Up	Shifted	CURSOR	(Reverse Shifted Q)
Cursor Right		CURSOR	(Reverse])
Cursor Left	Shifted	CURSOR	(Reverse Shifted])

Then use the CURSOR LEFT key to move the cursor back to the Y.

```
PRINT "MY PET BYTES"■ ←Press CURSOR LEFT
PRINT "MY PET BYTES"← Press CURSOR LEFT
PRINT "MY PET BYTES"← Press CURSOR LEFT
PRINT "MY PET BYTES"← Press CURSOR LEFT
PRINT "MY PET BYTES"← Press CURSOR LEFT
PRINT "MY PET BYTES"
```

Type over the Y with an I, and press the RETURN key to exit the statement. The corrected string is displayed on the following line:

```
PRINT "MY PET BYTES"← Type I over Y
PRINT "MY PET BITES"← Press RETURN
MY PET BITES ←——————— PRINT statement displayed on screen

READY.
■
```

An alternative method when editing text within quotation marks is to press the RETURN key, which removes the cursor from the current display line. This places the cursor in the first character position of the next line. (The computer may respond in various ways, depending on the statement. Do not concern yourself with the computer response just yet — your task is to edit the string text.) Move the cursor up to the line using CURSOR UP, and move the cursor left to the character to be edited.

Type in this statement:

```
PRINT "MY PET BYTES"
```

Press the RETURN key to drop the cursor down to a lower line.

```
PRINT "MY PET BYTES"← Press RETURN key
MY PET BYTES

READY.
■
```

Ignoring the screen response, press the CURSOR UP key (shifted) repeatedly to move the cursor up to the original statement line.

```
CURSOR UP key pressed → PRINT "MY PET BYTES"
CURSOR UP key pressed → MY PET BYTES
CURSOR UP key pressed →
CURSOR UP key pressed → READY.
                        ■
```

Press the CURSOR RIGHT key several times to move the cursor rightward to the Y character. Type over the Y character with an I.

```
PRINT "MY PET BYTES"
PRINT "MY PET BITES"← Y typed over with I character
```

Press the RETURN key. Upon pressing the RETURN key the statement is executed. The string is displayed with the new I character replacing the corrected I.

```
PRINT "MY PET BITES"← Type I over Y
MY PET BITES ←——————— RETURN key pressed

READY.
■
```

Like the CURSOR LEFT key, the CURSOR RIGHT and the CURSOR UP/DOWN become part of the string text when entered following an odd-numbered set of quotation marks.

But the INSERT/DELETE keys respond a little differently.

The INSERT key, when entered following an odd-numbered set of quotation marks, becomes a character in the text string. It is represented on the screen by the INSERT symbol (◫). But when the PRINT statement is executed and the string is displayed, the INSERT character has no effect on the display.

The DELETE key is the only cursor key not affected by the presence of quotation marks. To edit our example statement with the DELETE key, simply press the DELETE key to delete text back to the position which must be edited.

```
PRINT "MY PET BYTES"▒ ←—Press DELETE key
PRINT "MY PET BYTES▒ ←—Press DELETE key
PRINT "MY PET BYTE▒ ←—Press DELETE key
PRINT "MY PET BYT▒ ←—Press DELETE key
PRINT "MY PET BY▒ ←—Press DELETE key
PRINT "MY PET B▒
```

EDITING PROGRAM STATEMENTS

Do not read this section until you understand programming.

This section explains how to duplicate and edit similar BASIC statements using line numbers.

Line Duplication

Many programs have several similar or identical statements. It is often efficient to duplicate several statements from one original statement, rather than re-entering the statement many times. On the CBM computer, each program statement must be assigned a unique line number. **By changing the line number of a statement you can create a new statement without erasing the original.**

Enter the following program statement:

```
10 PRINT "*"
```

Suppose we need to enter five more identical statements, just like the one above. We could type in the statement five times, assigning each statement a unique line number. Or we could duplicate the statement five times by changing the line number of the original statement five times, as described below.

Enter the program statement to be duplicated. (If the program statement is already entered, list it using the LIST command, specifying the line number.) Press the CURSOR UP key until the cursor is positioned at the start of the line number. To change the statement line number, type over the number 10 with a new line number:

```
20▒PRINT "*" ←— Type over line number 10 with line number 20
```

When the new line number has been entered, press the RETURN key to create a new statement from the original. The RETURN key must be pressed after each line number change. If you list the program, both statements are displayed:

```
 20 PRINT "*"
LIST

 10 PRINT "*"
 20 PRINT "*"
READY.
```

Editing Similar Program Statements

Editing similar program statements follows the same procedure described in duplicating program statements. LIST the statement to be duplicated, specifying its line number. Move the cursor up to the program statement and type over the line number with a new line number. Then using the CURSOR RIGHT, INSERT or DELETE keys, move the cursor to edit the statement as needed. Press the RETURN key, and the new statement created from the original statement is sent to memory. Do not worry if the original statement is not displayed on the screen. It is still in memory and will be displayed if you LIST the program.

Ultimately we want to create a short example program that looks like this:

```
10 PRINT "*"
20 PRINT " *"
30 PRINT "  *"
40 PRINT "   *"
50 PRINT "    *"
```

Because all five lines are similar, we will duplicate and edit the first statement four times to avoid redundant typing of identical text.

Enter program statement 10 and press the RETURN key. To create the program statement on 20, move the cursor up to line number 10 and type over the 10 with a 20. Press the CURSOR LEFT key repeatedly until the cursor is positioned on top of the asterisk (*). Press the INSERT key once to move the asterisk over one space to the right, leaving a blank space at the current cursor position. Press the RETURN key to send the new program statement to memory. Although line number 10 is not displayed on the screen, it is still in computer memory, as shown by a program LIST:

```
 20 PRINT " *"
LIST

 10 PRINT "*"
 20 PRINT " *"
READY.
```

Statement 20 created from statement 10

Statements 30, 40, and 50 are created in the same manner as statement 20 by moving the cursor up to the statement, then editing the line number and text. After entering all five statements, the screen should look like this:

```
50 PRINT "    *"
```

Type in LIST to display the entire program:

```
50 PRINT "    *"
LIST

 10 PRINT "*"
 20 PRINT " *"
 30 PRINT "  *"
 40 PRINT "   *"
 50 PRINT "    *"
READY.
```

As you can see, duplicating and editing similar program statements is an efficient means of entering similar statements.

BASIC 4.0 SCREEN EDITING EXTENSIONS

BASIC 4.0 has a number of screen editing capabilities not available with earlier releases of CBM BASIC. These added capabilities are generally used within programs; they are not used to edit screen data in immediate mode, therefore they are described in Chapter 5.