

The final installment of a four-part programming tutorial

BASIC for Beginners

Lesson IV

By David W. Ostler

In "BASIC for Beginners: Lesson III" [February 1988, Page 20], I left you with a programming exercise. There are many solutions. Those of you who want to see what one possible solution looks like, just examine the listing of DATABASE, which accompanied "Lesson III."

Well, this is the final installment in the series. I hope I've helped you to develop your programming skills, or have helped you to better understand how commands work. Again I stress that we have not covered all the commands the Color Computer uses, but we have touched on many of the more common commands. To learn more about programming, keep reading *THE RAINBOW* and studying the manuals that accompany your computer.

This month we put the finishing touches on our database program, adding disk and tape I/O and enhancing some of the program's features.

Because the program has become rather intricate, I've included a line-by-line breakdown of this month's additions. (For a line-by-line description of the rest of the program, see Page 24 of February's "Lesson III.")

The best way to understand and plan a complicated program is to break it into blocks, identifying them by REM statements. Or you can mark the blocks' functions on a hard copy of the program. Use these methods to write your programs and keep the program functions straight in your mind.

OPEN

The OPEN command opens up a file on disk, tape or RAM memory for input/output (I/O) operations. This command can also name the file, establish length and define the device to be used.

Proper syntax for this command is OPEN"X\$",Y,A\$, where X\$ is either I

Dave Ostler is an IC layout designer and the systems manager for a CAD main-frame system. He teaches CAD and electronics at Guilford Technical Community College. Dave is married and has three children, Avis, Chuck and Erik.

or \square , for input or output, respectively. Υ is the device to be opened, and $A\$\$$ is the filename of the file to be accessed. An example of the use of the OPEN command is OPEN "I", 1, DATA/PRG, which opens a file named DATA/PRG for loading of data into the computer. The device used for "input from" is the disk drive. Note that I represents *input from* and \square represents *output to*.

The following are device specifiers, which should be used in place of the Υ

for determining the device for I/O "to" or "from":

Device	Specifier
Disk	1
Tape	-1
Printer	-2
Keyboard	0

EOF

The EOF command detects when an "End Of File" has been reached, and is

always used in an IF/THEN format.

The proper syntax for this command is IF EOF (x) THEN $xxxx$, or IF EOF (x) THEN CLOSE, where x is the End Of File indicator and $xxxx$ is the line number in the program where the program should go if an End Of File has been detected. Note that the command line reads, "If End Of File is x then go to $xxxx$, or if End Of File is x then close file." The End Of File indicators are listed as follows. These numbers indi-

Line	Description	Line	Description
500	sets N equal to zero, clears the screen and allows the input of string variable $A\$\$$, which is the name of the file to be used for data I/O.	580	properly closes the file that was opened in Line 567.
502	tests to see if there has been an entry for string variable $A\$\$$; if there has been no entry then the program will return to Line 2000.	590	forces a jump to Line 900.
505-520	test the value of variable A to see whether the system is tape or disk.	600	clears the screen and allows the entry of string variable $A\$\$$.
540	a remarked line.	602	sets a new value for $A\$\$$ equal to the old value of $A\$\$$ entered in Line 600. This is done to avoid any file errors that may occur due to too many characters being used in the filename. Remember that your computer recognizes only filenames of a maximum eight characters long with a three-character extension. The extension is a label placed on the program name that tells the computer what type of program is being loaded or saved. See your manuals for more information on filenames and extensions.
545	opens an input file to the cassette with the name found in $A\$\$$, which was entered in Line 500.	605-620	test the value of variable A to see whether the system is tape or disk.
550	checks to see if the End Of File has been reached. If it detects an EOF condition, it will jump to Line 560. If no EOF condition has been detected, the program will then go to the next line, which is Line 552.	650	a remarked line.
552	inputs data from the cassette drive. Please note that this data has dimensioned variable labels, each variable related to the dimension variable label N.	655	opens an output file to the cassette drive with the name found in $A\$\$$.
553	adds one count to the dimension variable label N. The computer does not care whether the data comes from the keyboard, tape drive or disk drive — it treats all the data the same way.	660	sets a FOR/NEXT command with the value of variable Z to be set from zero to the value of N-1. (The reason for this is that if we left the value for zero to N, we would have a blank file left at the end of the data file when we saved the data file to disk or tape. This would use up data space and not allow us to make the most of our data space available.
554	forces a jump to Line 550 to test for an EOF condition.	661	prints data to the cassette drive. These variables are printed as they relate to the variable label Z. The program then adds 1 to the value of Z and causes the program to go back to Line 660 to start the next count of the FOR/NEXT loop set up in Line 660. When the count conditions are satisfied for the FOR/NEXT loop, the line will then properly close the file opened in Line 655 and force a jump to Line 2000, which is the menu selection area.
555	properly closes the file that was opened in Line 545.	675	a remarked line.
560	forces a jump to Line 900.	680	opens an output file to the disk drive with the name found in $A\$\$$.
565	a remarked line.	682	sets a FOR/NEXT command with the value of Z to be set from 0 to the value of N-1.
567	opens an input file to the disk drive with the name found in $A\$\$$.	684	prints data to the disk drive.
568	checks to see if the End Of File has been reached. If it detects an EOF condition, it will jump to Line 580. If no EOF condition has been detected, the program will then go to the next line, which is Line 572.		
572	inputs data from the disk drive. This data has dimensioned variable labels, each variable related to the dimension variable label N.		
573	adds one count to the dimension variable label N.		
579	forces a jump to Line 568 to test for an EOF condition.		

cate that an End Of File condition has been detected:

Device	Indicator
Disk	1
Tape	-1

CLOSE

The CLOSE command properly terminates any file that was previously

opened. Failure to properly terminate open files will result in an FD Error (bad file data). These files may or may not be recoverable; usually they are lost and all data contained in them is lost, as well. Sometimes you can recover it by repairing the files or by using a disk repair utility to look at the data and print it out.

If for any reason you interrupt a program that has previously opened a file, or you suspect a file has been left

open, type CLOSE and press ENTER. This command will then properly terminate any file that might have been left open. This can be done after a break or error; be sure to enter the command *before* a reset, cold poke or warm start has occurred.

Proper syntax for this command is CLOSE, CLOSE x , where x is the proper device specifier or any other form that adds the proper device specifier on the end of the command.

Line	Description	Line	Description
685	adds 1 to the value of Z and causes the program to go back to Line 682 to start the next count of the FOR/NEXT loop set up in Line 682. When the count conditions are satisfied for the FOR/NEXT loop, it goes on to Line 686.	810-831	perform the same operations for entering and changing variables C\$, D\$ and E\$, and then force a jump to Line 700.
686	properly closes the file opened in Line 680.	900	sets up a FOR/NEXT value for B. (As explained for lines 660 and 682, when displaying or printing variables entered in arrays, you must use the variable used in counting the array, which, in this case, is N minus 1. This counts down the variable. To fail to do so will cause a blank file to be displayed.
687	forces a jump to Line 2000.	910-940	display text with associated variables as they relate to B.
700-770	print out the string variables B\$, C\$, D\$ and E\$, and allow you to select which one to correct.	950	prints text at the specified location.
880	clears the screen and allows the entry of B\$ as it is related to Z. It will be used for a new value for B\$ as related to N.	960	sets I\$ to an INKEY\$ function and will continue to Line 970 if any key is pressed.
801	sets B\$(N) equal to the same character values as B\$(Z); forces a jump to Line 700.		

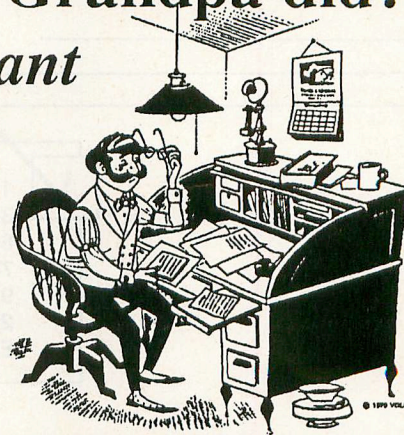
Still keeping the books the way Grandpa did? Then you need CoCo-Accountant

Tired of scrounging through old shoeboxes full of receipts, canceled checks and bills? Looking for an easy way to organize your finances when the tax man calls? Then you need **CoCo-Accountant**, the best-selling home and small business accounting program for the Color Computer. All you have to do is set up a chart of accounts and begin entering transactions. Checks, credit card expenses, income. In any order. **Just toss it in and CoCo-Accountant sorts it out.** No fuss, no muss, no mess.

When you're through, CoCo-Accountant will dazzle you with an array of reports that will answer the three basic questions we all ask about our finances: **Where did it come from? Where did it go? And what can I deduct from my taxes?**

Here's what it can do for you:

- ⇒ List and total expenses and income by **month**.
- ⇒ List and total expenses and income by **account**, for any month or the whole year.
- ⇒ List and total expenses or income by **payee or income source** for any month or the whole year.
- ⇒ Track, list and summarize **tax-deductible expenses**.
- ⇒ Track, list and summarize expenses subject to **sales tax**. Even calculates total sales tax you paid!
- ⇒ Produce a **printed spreadsheet** showing activity by month and account for the whole year!
- ⇒ **Balance your checkbook**, of course!
- ⇒ Sort entries by date and store files on tape or disk.
- ⇒ Up to 900 entries in a single file.
- ⇒ **Requires 64K CoCo or Coco 3.**



Coco-Accountant is \$34.95 on tape or disk. Be sure to specify which you want when you order. We accept VISA and MasterCard. COD orders, add \$3.00. Send check or money order to the address below or call our toll-free order line. For information, call 301-521-4886.

Federal Hill Software 8134 Scotts Level Rd. Baltimore, Md. 21208. Toll-free orders 800-628-2828 Ext. 850

PRINT#

The PRINT#x command prints the characters following this command to the device specified. The device specifiers used for the PRINT#x command are the same ones used for the OPEN command.

Proper syntax for this command is PRINT#-1 for the cassette drive, PRINT#-2 for the printer, PRINT#1 for the disk drive, or PRINT#A\$, where A\$ is the proper device specifier as determined in the program.

LEFT\$

This command allows you to select the left string of characters within a specified string of characters. It is useful in various text data manipulations.

The proper syntax for this command is B\$ = LEFT\$(A\$,x), where B\$ is the new character string to be defined, A\$ is the text string to be manipulated and x equals the number of characters to use in the length of the string manipulation (a number from 1 to 255). A common practice is to use the same string variable for the manipulation. This can be accomplished by the command A\$ = LEFT\$(A\$,B), which reads, "String variable A\$ has a new value of the old value of A\$, but only the first eight characters on the left of the old value of A\$."

Block	Lines	Label
1	0-85	Setup
3	95-220	Data Input
3A	300-330	Entry Error Correction
5	500-520	Output Device Setup
5A	540-560	Cassette Input
5B	565-590	Disk Input
6	600-620	Input Device Setup
6A	650-661	Cassette Output
6B	675-687	Disk Output
4A	700-831	Editor
4	900-980	Text Display
1A	1000-1200	System Type Display
2	2000-2010	Main Menu
7	5000	Program End Routine
3B	6000	Maximum File Size Routine

Figure 1

We will not go into the other string manipulation commands.

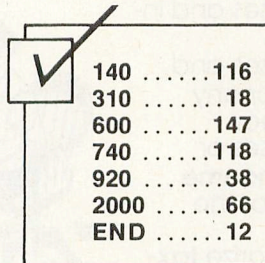
The Program

I mentioned earlier the method of "blocking off" your program, in which you take a hard copy of it and physically draw lines between distinct routines. You then label each routine, explaining its function. This makes the program much more understandable to you and to anyone else who looks at the listing. I blocked off the final version of DATABASE as shown in Figure 1.

Well, I guess this is it. We have cov-

ered many of BASIC's commands and have developed a small database program. The database is rather crude; I left it that way so you could practice the skills you have learned in this series. You can use what you have learned to enhance the database program, perhaps adding a feature that allows adding more data to an existing database, increasing the array sizes, etc.

(Questions or comments regarding this tutorial may be directed to the author at 901 Ferndale Blvd., High Point, NC 27260. Please enclose an SASE when writing for a reply.) □



140	116
310	18
600	147
740	118
920	38
2000	66
END	12

The listing: DATABASE

```
Ø 'BASIC NAME DATABASE PROGRAM.
THIS PROGRAM IS TO BE USED WITH
THE BASIC PROGRAMMING COURSE
WRITTEN BY DAVID W. OSTLER, COPY
RIGHT 1987
1Ø CLEAR1ØØØ:T=1ØØ:N=Ø:DIMB$(1Ø)
:DIMC$(1Ø):DIMD$(1Ø):DIME$(1Ø)
2Ø 'CHECK FOR TAPE OR DISK SYSTE
M
3Ø A=PEEK(188)
4Ø 'CHECK FOR 16K OR 64K SYSTEM
5Ø B=PEEK(116)
6Ø IF(A=14 AND B=127) THEN GOSUB
```

```
1ØØØ
7Ø IF(A=6 AND B=127) THEN GOSUB1
Ø1Ø
8Ø IF(A=6 AND B=63) THEN GOSUB1Ø
2Ø
85 GOTO2ØØØ
95 'FILES ENTERED HERE
1ØØ CLS:PRINT"ADDRESS DATABASE #
OF FILES";N:PRINT:LINEINPUT"ENT
ER NAME
";B$(N)
11Ø PRINT:LINEINPUT"ENTER ADDRES
S
";C$(N)
12Ø PRINT:LINEINPUT"ENTER CITY,
ST, &ZIP ";D$(N)
13Ø PRINT:LINEINPUT"ENTER TELEPH
ONE NO.
";E$(N)
14Ø CLS:PRINT:PRINT"1. NAME- ";B
$(N)
15Ø PRINT:PRINT"2. STREET-";C$(N
)
16Ø PRINT:PRINT"3. STATE- ";D$(N
)
```



```

170 PRINT:PRINT"4. PHONE- ";E$(N
)
180 PRINT@357,"PRESS <C> TO CONT
INUE":PRINT@399,"OR":PRINT@416,"
PRESS THE NUMBER TO CORRECT"
190 I$=INKEY$:IFI$=""THEN190ELSE
IFI$="1"THEN300ELSEIFI$="2"THEN3
10ELSEIFI$="3"THEN320ELSEIFI$="4
"THEN330ELSEIFI$="C"THEN200ELSE1
90
200 N=N+1:IFN=11GOTO60000
210 CLS:PRINT@456,"ANOTHER ENTRY
(Y/N)"
220 I$=INKEY$:IFI$=""THEN220ELSE
IFI$="Y"THEN100ELSEIFI$="N"THEN9
00ELSE220
300 CLS:PRINT:PRINT:LINEINPUT"EN
TER NAME
";B$(N):GOTO140
310 CLS:PRINT:PRINT:LINEINPUT"EN
TER ADDRESS
";C$(N):GOTO140
320 CLS:PRINT:LINEINPUT"3. STATE
- ";D$(N):GOTO140
330 CLS:PRINT:PRINT:LINEINPUT"EN
TER TELEPHONE NO.
";E$(N):GOTO140
500 N=0:CLS:INPUT"ENTER NAME OF
FILE TO BE LOADED";A$
502 IFA$=""THEN20000
505 'CHECK FOR DISK OR TAPE
510 IFA=6THEN545
520 IFA=14THEN565
540 'CASSETTE TAPE READ IN
545 OPEN"I",-1,A$
550 IF EOF (-1) THEN 560
552 INPUT#-1,B$(N),C$(N),D$(N),E
$(N)
553 N=N+1
554 GOTO550
555 CLOSE
560 GOTO9000
565 'DISK SYSTEM READ IN
567 OPEN"I",1,A$
568 IF EOF (1) THEN 580
572 INPUT#1,B$(N),C$(N),D$(N),E$
(N)
573 N=N+1
579 GOTO568
580 CLOSE
590 GOTO9000
600 CLS:INPUT"ENTER NAME OF FILE
TO BE SAVED";A$
602 A$=LEFT$(A$,8)
605 'CHECK FOR DISK OR TAPE
610 IFA=6THEN650
620 IFA=14THEN675
650 'CASSETTE TAPE SAVE
655 OPEN"O",-1,A$
660 FORZ=0 TO N-1
661 PRINT#-1,B$(Z);",";C$(Z);",
";D$(Z);",";E$(Z);",";
662 NEXTZ

```

Coco Graphics Designer

Only \$29.95

The Coco Graphics Designer produces beautiful Greeting Cards, Banners, and Signs for holidays, birthdays and other occasions.

The program features picture, border, and character font editors, so that you can modify or expand the already built in libraries. Plus a special "grabber" utility is included to capture areas of high resolution screens for your picture library.

Requirements: a Coco I, II or III with at least 32K, one disk drive, BASIC 1.0/1.1, ADOS 1.0/1.1 or JDOS. Printers supported include: Epson RX/FX, Gemini 10X, SG10, NX10, C-Itoh 8510, DMP

100/105/110/130/430 CGP220, many Okidata (check with Zebra), Seikosha GP100/250, Gorilla Banana, Legend 808.
#C323 Coco Graphics Designer

Picture Disk #1

This supplementary picture library diskette contains over one hundred additional pictures.

#C333 Picture Disk #1 \$14.95

Colored Paper Packs

150 sheets (50 each red, yellow, blue) with 60 matching envelopes. Perfect for making your productions outstanding.

#C274 Paper Pack \$19.95

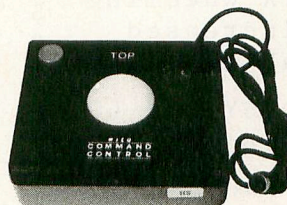


It's fun making your own Greeting Cards, Signs, and Banners with Zebra's Coco Graphics Designer.

WICO TRACKBALL

Only \$29.95

Order Cat#TBCC



WICO designed these trackballs specifically for the Radio Shack Color Computer joystick port. Features 360-degree movement and quick-action fire button for smooth, arcade response and feel. Works great with Coco joystick and mouse software.

Ordering Instructions: All orders add \$3.00 Shipping & Handling. UPS COD add \$3.00. VISA/MC Accepted. NY residents add sales tax.



The Car Sign Designer program enables you to easily create distinctive bright yellow diamond shaped car signs. Everything you need is provided including two reusable clear plastic sign holders with suction cups, 50 sheets of bright yellow fanfold paper, and the Car Sign Designer program disk and instructions. ... Hardware, DOS, and printer requirements are the same as for our CoCo Graphics Designer above. Order Cat#CSCC, \$29.95. For six additional sign holders, order Cat#CS6PK \$9.95.

Zebra Systems, Inc.

78-06 Jamaica Avenue
Woodhaven, NY 11421
(718) 296-2385


```

663 CLOSE
664 GOTO2000
675 'DISK SYSTEM SAVE
680 OPEN"O",1,A$
682 FORZ=0 TO N-1
684 PRINT#1,B$(Z);",",C$(Z);",",
D$(Z);",",E$(Z);",",
685 NEXTZ
686 CLOSE
687 GOTO2000
700 CLS:PRINT:PRINT"1. NAME- ";B
$(Z)
710 PRINT:PRINT"2. STREET-";C$(Z
)
720 PRINT:PRINT"3. STATE- ";D$(Z
)
740 PRINT:PRINT"4. PHONE- ";E$(Z
)
750 PRINT@357,"PRESS <C> TO CONT
INUE":PRINT@399,"OR":PRINT@416,"
PRESS THE NUMBER TO CORRECT"
770 I$=INKEY$:IFI$=""THEN770ELSE
IFI$="1"THEN800ELSEIFI$="2"THEN8
10ELSEIFI$="3"THEN820ELSEIFI$="4
"THEN830ELSEIFI$="C"THEN970ELSE7
70
800 CLS:PRINT:PRINT:LINEINPUT"EN
TER NAME
";B$(Z)

```

```

801 B$(N)=B$(Z):GOTO700
810 CLS:PRINT:PRINT:LINEINPUT"EN
TER ADDRESS
";C$(Z)
811 C$(N)=C$(Z):GOTO700
820 CLS:PRINT:PRINT:LINEINPUT"3. STATE
- ";D$(Z)
821 D$(N)=D$(Z):GOTO700
830 CLS:PRINT:PRINT:LINEINPUT"EN
TER TELEPHONE NO.
";E$(Z)
831 E$(N)=E$(Z):GOTO700
900 FORZ=0 TO N-1
910 CLS:PRINT:PRINT"1. NAME- ";B
$(Z)
920 PRINT:PRINT"2. STREET-";C$(Z
)
930 PRINT:PRINT"3. STATE- ";D$(Z
)
940 PRINT:PRINT"4. PHONE- ";E$(Z
)
950 PRINT@320,"PRESS [C] TO
CONTINUE, [R] TO RETURN T
O MAIN MENU,":PRINT@399,"OR":PRI
NT@422,"[E] TO EDIT ENTRY"
960 I$=INKEY$:IFI$=""THEN960ELSE
IFI$="C"THEN970ELSEIFI$="R"THEN2
000ELSEIFI$="E"THEN700ELSE960
970 NEXTZ
980 GOTO2000
1000 CLS:PRINT@230,"32/64K DISK
SYSTEM";FORX=1TO1000STEP1:NEXT
X:RETURN
1010 CLS:PRINT@228,"32/64K CASS
ETTE SYSTEM";FORX=1TO1000STEP1:
NEXTX:RETURN
1020 CLS:PRINT@229,"16K CASSETT
E SYSTEM";FORX=1TO1000STEP1:NEX
TX:RETURN
2000 CLS:PRINT@32,"WELCOME TO
THE BASIC DATABASE ";:PRINT@71,"
WOULD YOU LIKE TO:"::PRINT@133,"
B)EGIN A NEW DATABASE";:PRINT@19
7,"A)BORT THIS PROGRAM";
2005 PRINT@261,"L)OAD A NEW DATA
BASE";:PRINT@325,"S)AVE THIS DAT
ABASE";:PRINT@389,"V)IEW THE DAT
ABASE";:PRINT@458,"[SELECT ONE]"
;
2010 I$=INKEY$:IFI$=""THEN2010EL
SEIFI$="B"THEN950ELSEIFI$="A"THEN
5000ELSEIFI$="L"THEN5000ELSEIFI$=
"S"THEN6000ELSEIFI$="V"THEN9000EL
S2010
5000 CLS3:PRINT@224,"REBO
OTING TO BASIC":SOUND200,2:SOUND
100,3:FORX=1TO1000STEP1:NEXTX:CL
S:END
6000 CLS:PRINT@224,"MAXIMUM
FILE SIZE REACHED":SOUND200,2:SO
UND100,3:FORX=1TO1000STEP1:NEXTX
:GOTO900

```

```

$ $ $ $ $ $ $ $ $ $ $ $ $ $ $ $ $
$ YOU could WIN BIG with $
$ LOT-PRO! $
$ $
$ HANDICAPS ANY LOTTO! $
$ Any Country, Any State $
$ LOT-PRO is specifically $
$ designed for the Color $
$ Computer and features: $
$ -A choice of SIX wheeling $
$ systems for increased $
$ winning probabilities $
$ -Unique LOT-PRO SYSTEM 60 $
$ number selection routine $
$ ONLY $25.95 (disk or tape) $
$ CJN ENTERPRISES $
$ P.O. Box 40487 $
$ Bakersfield, CA. 93384 $
$ (805)-836-1323 $
$ California residents $
$ must add 6% sales tax. $
$ $
$ Invest in LOT-PRO! $
$ It could make YOU $
$ RICH! $
$ $ $ $ $ $ $ $ $ $ $ $ $ $ $ $

```

