

DECEMBER 1981 Vol. 6, No. 12

\$2.50 in USA/\$2.95 in Canada

A McGraw-Hill Publication

BYTE

the small systems journal



HEX ANGELS

PLAYER 0	0 0 2 4 0
PLAYER 1	0 2 1 3 0
PLAYER 2	0 1 5 6 0
PLAYER 3	0 2 1 0 0

YOUR SCORE
1 4 5 2 0

P	0 1 0 0 0 1 1 0 1 1 0 0 0 0 1
C	
S	0 1 1 0 1 1 0 1
P	0 1 1 0 0 1 0 1
A	0 1 1 1 0 0 1 1



COMPUTER GAMES

INSERT 25 SECTOR DISK →

Generating Programs Automatically

Let Your Apple II Do the Programming

Jacob R Jacobs
1903 Fordham Way
Mountain View CA 94040

Wouldn't it be great if your computer could write programs? Or if it could write those portions of your programs that you find most tedious? With the three utility programs described in this article, you simply answer a few questions interactively, and the computer automatically generates the Applesoft BASIC program for you.

The three programs are written in Applesoft BASIC, but they can be easily modified to run in, and generate programs for, another version of BASIC. The utility programs generate BASIC programs for these three sections:

- Data entry section: the area where repetitive prompting, input, and range checking are performed.

- Data output section: the part of

your program that requires a careful determination of the tabs for printing headings and for printing the data in columns where the first or last character or decimal point lines up.

- Instruction section: most programs begin with instructions on how to use them, or provide some introductory text. You must be careful that the text doesn't wrap on the screen in the middle of words. It is also time consuming to center headings.

To create a program using these utilities, simply run the utility program and answer the questions. When you are finished, the utility will generate a BASIC program and store it in a text file. To use the text file, just EXEC it into your program.

Listing 1a shows a sample dialog

Listing 1: Products of the CREATE INPUT program. Listing 1a shows the sample dialog (the user's inputs are indicated in lowercase), while listing 1b shows the program generated in response to CREATE INPUT's queries.

1a

```
HOW MANY VARIABLES? 3
DIMENSION OF ARRAYS? 20
NAME OF VARIABLE 1 ($ FOR STRING)
?item$
PROMPT LINE FOR ITEM:
?enter product description
NAME OF VARIABLE 2 ($ FOR STRING)
?pr
PROMPT LINE FOR PR:
?unit price
DO YOU WANT A RANGE CHECK (Y/N)? y
MINIMUM ACCEPTABLE VALUE? 0
```

```
MAXIMUM ACCEPTABLE VALUE? 10000
NAME OF VARIABLE 3 ($ FOR STRING)
?qn
PROMPT LINE FOR QN:
?quantity
DO YOU WANT A RANGE CHECK (Y/N)? y
MINIMUM ACCEPTABLE VALUE? 1
MAXIMUM ACCEPTABLE VALUE? 144
VAR. INDEX FOR TERMINATION? 1
WHAT IS THE TERMINATING VALUE? end
STARTING PROGRAM LINE? 1000
INCREMENT FOR PROGRAM? 10
```

for the input program. Assume that you want to enter a product name, price, and quantity, and then print out a formatted invoice that shows quantity, product name, price, extended price, and total. These utilities will help you write the program, but they won't do the entire job. You must fill in the middle, and modify the automatically generated programs where necessary.

First, run the CREATE INPUT program. After it has finished, a BASIC program will be generated and displayed on the screen. You will be asked if you want to save this program on the disk, and if so, under what name. Listing 1b shows the program that results from this dialog.

You are also asked to indicate the number of variables you are using, in this case three: ITEM\$, PR, and QN. You are then asked to provide the dimensions of the arrays that these variables will require. In this example we will have not more than 20 items on an invoice. Note that you are asked if you want range checks for numeric data only, not for string data such as ITEM\$.

1b

```
1000 DIM ITEM$(20),PR(20),QN(20)
1010 I=1
1020 PRINT "ENTRY ";I
1030 INPUT "ENTER PRODUCT DESCRIPTION
";ITEM$(I)
1040 IF ITEM$(I)="END" GOTO 1100
1050 INPUT "UNIT PRICE ";PR(I)
1060 IF PR(I)<0 OR PR(I)>10000 GOTO 1050
1070 INPUT "QUANTITY ";QN(I)
1080 IF QN(I)<1 OR QN(I)>144 GOTO 1070
1090 I=I+1
: GOTO 1020
```


Listing 2: Sample dialog from the CREATE OUTPUT program.

```

HOW MANY VARIABLES? 4
NAME OF VARIABLE 1 ($ FOR STRING)
? Q1
WIDTH OF FIELD? 4
DECIMAL DIGITS? 0
HEADING 1? QUAN
HEADING 2?
HEADING 3? -----
NAME OF VARIABLE 2 ($ FOR STRING)
? ITEM$
WIDTH OF FIELD? 12
HEADING 1? PRODUCT
HEADING 2? DESCRIPTION
HEADING 3? -----
NAME OF VARIABLE 3 ($ FOR STRING)
? PR

WIDTH OF FIELD? 8
DECIMAL DIGITS? 2
HEADING 1? UNIT
HEADING 2? PRICE
HEADING 3? -----
NAME OF VARIABLE 4 ($ FOR STRING)
? EP
WIDTH OF FIELD? 10
DECIMAL DIGITS? 2
HEADING 1? EXTENDED
HEADING 2? PRICE
HEADING 3? -----
STARTING PROGRAM LINE? 3000
INCREMENT FOR PROGRAM? 10
SPACE BETWEEN COLUMNS? 1
    
```

Listing 3: Sample dialog from the CREATE INSTR program.

```

APPROXIMATELY HOW MANY LINES? 20
TYPE 'CONTROL-Q' TO QUIT
ANSWER QUESTIONS WITH 'Y' OR 'N'
    
```

```

TYPE LINE 1
      INVOICE PROGRAM
TYPE LINE 2
    
```

```

TYPE LINE 3
THIS PROGRAM WILL PRINT AN INVOICE OR
TYPE LINE 4
PURCHASE ORDER FOR UP TO 20 ITEMS.
TYPE LINE 5
WHEN PROMPTED TYPE PRODUCT DESCRIPTION,
TYPE LINE 6
UNIT PRICE AND QUANTITY. TYPE 'END'
TYPE LINE 7
FOR PRODUCT DESCRIPTION WHEN DONE,
TYPE LINE 8
    
```

INVOICE PROGRAM

```

THIS PROGRAM WILL PRINT AN INVOICE OR
PURCHASE ORDER FOR UP TO 20 ITEMS.
WHEN PROMPTED TYPE PRODUCT DESCRIPTION,
UNIT PRICE AND QUANTITY. TYPE 'END'
FOR PRODUCT DESCRIPTION WHEN DONE.
    
```

```

DO YOU WANT TO CHANGE A LINE? Y
WHAT LINE? 1
      INVOICE PROGRAM
    
```

```

IS THIS THE RIGHT LINE? Y
TYPE LINE 1
      INVOICE PROGRAM

      INVOICE PROGRAM
    
```

```

THIS PROGRAM WILL PRINT AN INVOICE OR
PURCHASE ORDER FOR UP TO 20 ITEMS.
WHEN PROMPTED TYPE PRODUCT DESCRIPTION,
UNIT PRICE AND QUANTITY. TYPE 'END'
FOR PRODUCT DESCRIPTION WHEN DONE.
    
```

```

DO YOU WANT TO CHANGE A LINE? N
STARTING PROGRAM LINE? 10
INCREMENT FOR PROGRAM? 10
10?TAB(13);"INVOICE PROGRAM"
20?
30?"THIS PROGRAM WILL PRINT AN INVOICE OR"
40?"PURCHASE ORDER FOR UP TO 20 ITEMS."
50?"WHEN PROMPTED TYPE PRODUCT DESCRIPTION,"
60?"UNIT PRICE AND QUANTITY. TYPE 'END'"
70?"FOR PRODUCT DESCRIPTION WHEN DONE."
    
```

INTRODUCTORY OFFER!

Turn your Micro-Computer into a Mini-Computer. Try the world's #1 programming language-COBOL! Finally at a price you can afford and with no risk!

Introducing...

NPS-MICRO-COBOL

This is the Naval Post Graduate School Cobol that you've heard so much about. Designed to pass the stringent government Hypo-COBOL tests used by GSA in their Compiler Certification Program. This is the first public release of version 2.1. This is an elaborate ANSI-COBOL subset. Comes complete with users manual in DeLuxe three-ring binder.

- Perfect for learning COBOL.
- Perfect for teaching COBOL
- FREE sample programs included
- Runs in 24K
- Requires 8080, Z-80*, or 8085 and standard CP/M* system
- Provided on standard 8" disk or Northstar Double Density CP/M 5"

Only \$69.95!

FREE ALGOL INCLUDED!

FREE BONUS. All purchasers receive a FREE copy of NPS-ALGOL at no extra cost. A favorite language in Europe, ALGOL is the original structured language. Comes with FREE sample programs.

MONEY-BACK GUARANTEE. If you're not completely satisfied with this software. You may return it within fifteen days for any reason and get a full refund.

Send Check, Money Order or Credit Card information and order a copy today! Please add \$2.50 shipping and handling on all orders.

Credit Card buyers: For Extra Fast service call (415) 527-7730

Order from: The Software Review
704 Solano Avenue, Albany, CA 94706

Yes, I want to run COBOL on my system! Enclosed find \$69.95 plus \$2.50 shipping/handling (California residents please add appropriate sales tax). I will receive the NPS-COBOL system plus a FREE copy of NPS-ALGOL. I understand that I may return the software within 15 days if not completely satisfied for a full refund.

NAME _____

COMPANY _____

STREET _____

CITY _____

STATE _____ ZIP _____

AMOUNT ENCLOSED \$ _____

Disk size desired: 5" 8"

Check Enclosed VISA

UPS C.O.D. Mastercharge

Card number _____

Expiration Date _____

Signature _____

Check here for more information

CP/M is a trademark of Digital Research and Z-80 is a trademark of Zilog

Listing 4: The completed invoice recording program. Lines 5, 1100, 2000 through 2040, and 4000 through 4040, were added by the programmer. Lines 2500 through 2530 were generated by CREATE INSTR, as were lines 10 through 70 and line 4050. All other lines were generated automatically.

```

5 HOME
10 PRINT TAB(13);"INVOICE PROGRAM"
20 PRINT
30 PRINT "THIS PROGRAM WILL PRINT AN
  INVOICE OR"
40 PRINT "PURCHASE ORDER FOR UP TO 2
  0 ITEMS."
50 PRINT "WHEN PROMPTED TYPE PRODUCT
  DESCRIPTION,"
60 PRINT "UNIT PRICE AND QUANTITY.
  TYPE 'END'"
70 PRINT "FOR PRODUCT DESCRIPTION WH
  EN DONE."
80 PRINT
1000 DIM ITEMS(20),PR(20),QN(20)
1010 I=1
1020 PRINT "ENTRY ";I
1030 INPUT "ENTER PRODUCT DESCRIPTION
  ";ITEM$(I)
1040 IF ITEM$(I)="END" GOTO 1100
1050 INPUT "UNIT PRICE ";PR(I)
1060 IF PR(I)<0 OR PR(I)>10000 GOTO 10
  50
1070 INPUT "QUANTITY ";QN(I)
1080 IF QN(I)<1 OR QN(I)>144 GOTO 1070
1090 I=I+1
  : GOTO 1020
1100 I=I-1
2000 TT=0
2010 FOR N=1 TO M
2020 EP(N)=QN(N)*PR(N)
2030 TT=TT+EP(N)
2040 NEXT N
2500 PRINT TAB(4);"INVOICE FOR"
2510 PRINT TAB(14);"ACME COMPANY"
2520 PRINT TAB(14);"1234 MAIN STREET"
2530 PRINT TAB(14);"ANYWHERE, USA"
3000 PRINT
3010 PRINT TAB(2);"QUAN";
3020 PRINT TAB(9);"PRODUCT";
3030 PRINT TAB(22);"UNIT";
3040 PRINT TAB(30);"EXTENDED";
3050 PRINT
3060 PRINT TAB(4);"";
3070 PRINT TAB(7);"DESCRIPTION";
3080 PRINT TAB(21);"PRICE";
3090 PRINT TAB(31);"PRICE";
3100 PRINT
3110 PRINT TAB(2);"----";
3120 PRINT TAB(7);"-----";
3130 PRINT TAB(21);"-----";
3140 PRINT TAB(30);"-----";
3150 PRINT
3160 FOR I=1 TO M
3170 A=QN(I)
3180 W%=4
  : D%=0
3190 GOSUB 60000
3200 PRINT TAB(6-LEN(A$));A$;
3210 A$=ITEM$(I)
3220 PRINT TAB(19-LEN(A$));A$;
3230 A=PR(I)
3240 W%=8
  : D%=2
3250 GOSUB 60000
3260 PRINT TAB(28-LEN(A$));A$;
3270 A=EP(I)
3280 W%=10
  : D%=2
3290 GOSUB 60000
3300 PRINT TAB(39-LEN(A$));A$;
3310 PRINT
3320 NEXT I
4000 A=TT
4010 GOSUB 60000
4020 PRINT
4030 PRINT " TOTAL";TAB(39-LEN(A$));A
  $
4040 PRINT
4050 PRINT "PLEASE REMIT WITHIN 30 DAY
  S. THANK YOU"
5000 END
60000 A=INT(A*10^D%+.5)/(10^D%)
60010 A$=STR$(A)
60020 RETURN

```

In order to terminate the data-entry loop, you are asked to give the index of the variable on which to terminate. In this case you answer 1 (ie: the first variable, ITEMS\$). The terminating value is END, since you have no item called END. Finally, you are asked for the starting program line and increment. Since you will be pulling these program segments from text files by using the EXEC feature, you must be sure that the program ranges do not overlap.

You must write the substance of the program yourself. In line 1040 there is a GOTO target that does not exist. This will be the first line of your own program. It will set M=I-1; M now contains the number of items in the invoice. Here is the program you might add:

```

2000 TT = 0
2010 FOR N = 1 TO M
2020 EP(N) = QN(N)*PR(N)
2030 TT = TT + EP(N)
2040 NEXT N

```

TT is the running total. Next you run the CREATE OUTPUT program. This program calls a small subroutine, which is to be located at line 60000:

```

60000 A=INT(A*10^D%+.5)/(10^D%)
60010 A$=STR$(A)
60020 RETURN

```

This subroutine converts the numeric variable A to a string variable A\$. W% and D% are the width and number of decimal places, respectively. W% is not used in this version of

the subroutine.

The CREATE OUTPUT program asks for the names of the variables you are using. In this case, you would answer: QN, ITEM\$, PR, EP, since you want the data printed in a different order than it was input. You are asked to provide three lines of heading for each column. The heading widths cannot be larger than those specified in the WIDTH OF FIELD? question. The complete dialog is shown in listing 2. Note that you can also specify the space between columns.

The last program creates screens full of instructions for you. It is a simple-minded text editor that generates print statements with the proper tabs. After you type in the text (without the line numbers and PRINT symbol), you have a chance to change any lines that need correction. Since lines are not numbered, you have to guess which line number is in error. The program confirms the line by printing it before you are asked to replace it. No line or character insertions or deletions are permitted, but you can always edit the completed BASIC program by adding or deleting lines.

Listing 3 shows the dialog for creating the instructions for your invoice program. Listing 4 shows the completed program, including the subroutine at 60000. Lines 4000 through 4040 had to be added to print the total. Listing 5 is a sample run of the invoice program. The CREATE INSTRUCTIONS program has also been used to create the company heading (ACME COMPANY) on the invoice. Only some of the line numbers of the generated program had to be changed. The example in listing 3 does not show the creation of the invoice heading.

All of the programs work in essentially the same way. The variable PLC (Program Location Counter, a term borrowed from assemblers) is used to keep track of the statement number assigned to each created program step. In the INPUT and OUTPUT programs, each line is placed in the variable L\$(J), where J is the Jth line. Let's decompose statement 360 in the CREATE INPUT program.

Text continued on page 362

BYTE

LEGAL NOTICE

US POSTAL SERVICE
STATEMENT OF OWNERSHIP, MANAGEMENT
AND CIRCULATION
(Required by 39 U.S.C. 3685)

1. Title of publication: *BYTE*
2. Date of filing: *September 8, 1981*
3. Frequency of issue: *Monthly*
- 3A. Number of issues published annually: *12*
- 3B. Annual Subscription Price: *\$19.00*
4. Complete mailing address of known office of publication: *70 Main Street, Peterborough, N.H. 03458*
5. Complete mailing address of the headquarters or general business offices of the publishers: *1221 Avenue of the Americas, New York, N.Y. 10020*
6. Full names and complete mailing address of publisher, editor, and managing editor: Publisher, *Virginia Londoner, 70 Main Street, Peterborough, N.H. 03458*; Editor, *Christopher P. Morgan, 70 Main Street, Peterborough, N.H. 03458*; Managing Editor, *Mark Haas, 70 Main Street, Peterborough, N.H. 03458*
7. Owner: The owner is *McGraw-Hill, Inc., 1221 Avenue of the Americas, New York, N.Y. 10020*. Stockholders holding 1 percent or more of stock are: *Donald C. McGraw, Jr., Harold W. McGraw, Jr., John L. McGraw, William H. McGraw, June M. McBroome, Elizabeth McGraw Webster, all of 1221 Avenue of the Americas, New York, N.Y. 10020, Public Employees Retirement Board of Ohio, 277 East Town Street, Columbus, Ohio 43215*
8. Known bondholders, mortgagees, and other security holders owning or holding 1 percent or more of total amount of bonds, mortgages or other securities: *None*
9. Not applicable.
10. Extent and nature of circulation:

	Average No. Copies Each Issue During Preceding 12 Months	Actual No. Copies of Single Issue Published Nearest to Filing Date
A. Total No. Copies	205,958	241,500
B. Paid Circulation		
1. Sales through dealers and carriers, street vendors and counter sales	33,743	56,845
2. Mail Subscription	160,869	179,405
C. Total Paid Circulation	194,632	236,250
D. Free distribution by mail, carrier or other means samples, complimentary, and other free copies	5,084	3,121
E. Total distribution	199,716	239,371
F. Copies not distributed		
1. Office use, left over, unaccounted, spoiled after printing	3,896	2,179
2. Return from news agents	2,346	<i>None to date</i>
G. Total	205,958	241,500

11. I certify that the statements made by me above are correct and complete.

BYTE
Virginia Londoner, Publisher

Listing 5: Sample run of the invoice program of listing 4.

INVOICE PROGRAM

THIS PROGRAM WILL PRINT AN INVOICE OR PURCHASE ORDER FOR UP TO 20 ITEMS, WHEN PROMPTED TYPE PRODUCT DESCRIPTION, UNIT PRICE AND QUANTITY. TYPE 'END' FOR PRODUCT DESCRIPTION WHEN DONE.

```

ENTRY 1
ENTER PRODUCT DESCRIPTION DOG
UNIT PRICE 19.95
QUANTITY 5
ENTRY 2
ENTER PRODUCT DESCRIPTION CAT
UNIT PRICE 12.95
QUANTITY 1
ENTRY 3
ENTER PRODUCT DESCRIPTION ELEPHANT
UNIT PRICE 999.75
QUANTITY 3
ENTRY 4
ENTER PRODUCT DESCRIPTION END
    
```

INVOICE FOR

ACME COMPANY
1234 MAIN STREET
ANYWHERE, USA

QUAN	PRODUCT DESCRIPTION	UNIT PRICE	EXTENDED PRICE
5	DOG	19.95	99.75
1	CAT	12.95	12.95
3	ELEPHANT	999.75	2999.25
TOTAL			3111.95

PLEASE REMIT WITHIN 30 DAYS. THANK YOU

Listing 6: The program-generating utilities, CREATE INPUT, CREATE OUTPUT, and CREATE INSTR.

CREATE INPUT

```

10 INPUT "HOW MANY VARIABLES? ";N
20 INPUT "DIMENSION OF ARRAYS? ";M
30 FOR I=1 TO N
  : MODE(I)=0
  : NEXT
40 FOR I=1 TO N
50 PRINT "NAME OF VARIABLE ";I;" ($ FOR STRING)"
60 INPUT V$(I)
70 IF RIGHT$(V$(I),1)="$" THEN MODE(I)=3
80 PRINT "PROMPT LINE FOR ";V$(I);": "
90 INPUT P$(I)
100 IF MODE(I)=3 GOTO 160
110 INPUT "DO YOU WANT A RANGE CHECK (Y/N)? ";Z$
120 IF Z$<>"Y" THEN MODE(I)=1
  : GOTO 160
130 INPUT "MINIMUM ACCEPTABLE VALUE? ";LV$(I)
140 INPUT "MAXIMUM ACCEPTABLE VALUE? ";HV$(I)
150 MODE(I)=2
160 NEXT I
170 INPUT "VAR. INDEX FOR TERMINATION? ";T
180 INPUT "WHAT IS THE TERMINATING VALUE? ";TV$
190 INPUT "STARTING PROGRAM LINE? ";FR
200 INPUT "INCREMENT FOR PROGRAM? ";INC
210 DIM L$(5+3*N)
220 PLC=FR
  : J=1
230 L$(J)=STR$(PLC)+" DIM "
240 FOR I=1 TO N
    
```

Listing 6 continued on page 358

IPEX **NEW!**

**"ELF" Interface
Converts Your
Typewriter Into Printer**



- Finest print quality
- Easy installation
- Fits IBM Electric and Electronic Typewriters
- Models for all popular computers
- Call or write for more information, today!

- Low cost
- Quick delivery

We export to all countries.

IPEX INTERNATIONAL INC.
5115 Douglas Fir Rd.
Calabasas, CA 91302 U.S.A.
Tel: (213) 710-1444 TLX/TWK: 910 494 2100

Circle 202 on inquiry card.

HIT-80 INC.



E-Z-80 ENTRY KIT
A Z-80 Micro Computer with CPU: CTC:
PIO: Prom Programmer: Read Outs: Key Pad:
Onboard ROM & RAM: Wire Wrap: DC
Regulator: Manual & Instructions.

Kit Price \$390.00

P.O. BOX 501—KAUFMAN, TEXAS (USA)—75142
A Subsidiary of Patrick Computer Systems, Inc., manufacturer of the 14120 Integrated business computer.

Circle 455 on inquiry card.

NORTHSTAR USERS

Add our **SCAN** COMMAND to your copy of NORTHSTAR BASIC and you can instantly display or print:

- All lines that contain a selected variable
- All occurrences of GOTO or GOSUB statements to a particular line number
- All references to any program line number
- All occurrences of any character, number, line number, group of characters or anything that is contained in any basic line

ONLY \$29.90 ON DISK

Add our enhanced **RENUMBER** capability to your copy of NORTHSTAR BASIC and you will be able to:

- Renumber only selected line numbers and move the entire group to another area of your program
- Keep specific line number sequences associated with designated areas of a program
- Append program modules and renumber them within your program
- Move to subroutines out of selected lines and move the subroutines to a designated area of the program

ONLY \$29.90 ON DISK

Both programs will naturally read to any copy of NORTHSTAR BASIC. Your BASIC can be SINGLE DENSITY, DOUBLE DENSITY or QUAD DENSITY and be loaded at any address above DEWEX. These programs use machine language additions to NORTHSTAR BASIC. Also, your copy of BASIC is modified to include these new features; you can use the NEW BASIC and have a permanent copy with SCAN and RENUMBER. Complete documentation included on each disk.

Please add \$2.00 shipping charge for each program or order both programs and save \$4.00 plus shipping charges. Ship only \$65.00 each.

CALL OR ORDER TODAY! 805 584-8259

Order Today—Send check, money order or credit card data to:

ET SOFTWARE SERVICES
1072 Cassia Pass Road
Carpinteria, CA 93013
(805) 584-8259

Circle 149 on inquiry card.

Listing 6 continued:

```

250 L$(J)=L$(J)+V$(I)+"( "+STR$(M)+")", "
260 NEXT I
270 L$(J)=LEFT$(L$(J),LEN(L$(J))-1)
280 GOSUB 620
290 L$(J)=STR$(PLC)+" I=1"
300 GOSUB 620
310 LOOP=PLC
320 L$(J)=STR$(PLC)+" ?"+CHR$(34)+"ENTRY "+CHR$(34)+" "; I"
330 GOSUB 620
340 FOR I=1 TO M
350 ER=PLC
360 L$(J)=STR$(PLC)+" INPUT "+CHR$(34)+P$(I)+" "+CHR$(34)+" "; "+V$(I)+"(I)"
370 GOSUB 620
380 IF I<>T GOTO 440
390 DN=J
400 Q$=""
410 IF MODE(I)=3 THEN Q$=CHR$(34)
420 L$(J)=STR$(PLC)+" IF "+V$(I)+"(I)="+Q$+TV$(I)+Q$+" GOTO "
430 GOSUB 620
440 IF MODE(I)<>2 GOTO 470
450 L$(J)=STR$(PLC)+" IF "+V$(I)+"(I)<"+LV$(I)+" OR "+V$(I)+"(I)>"+HV$(I)+" G
OTO "+STR$(ER)
460 GOSUB 620
470 NEXT I
480 L$(J)=STR$(PLC)+" I=I+1:GOTO "+STR$(LOOP)
490 GOSUB 620
500 L$(DN)=L$(DN)+STR$(PLC)
510 PRINT
: PRINT
520 FOR K=1 TO J
: PRINT L$(K)
: NEXT
530 INPUT "DO YOU WANT TO SAVE ON DISK?";Z$
540 IF Z$<>"Y" THEN END
550 INPUT "TEXT FILE NAME? ";F$
560 D$=CHR$(4)
570 PRINT D$;"OPEN";F$
580 PRINT D$;"WRITE";F$
590 FOR K=1 TO J
: PRINT L$(K)
: NEXT K
600 PRINT D$;"CLOSE";F$
610 END
620 PLC=PLC+INC
: J=J+1
: RETURN

```

CREATE OUTPUT

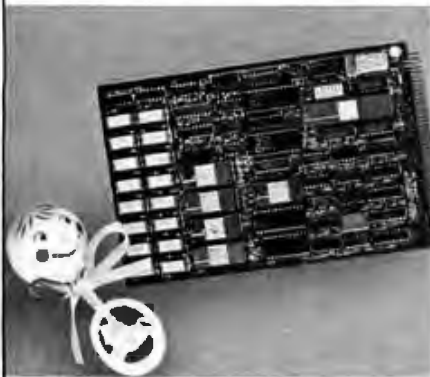
```

10 INPUT "HOW MANY VARIABLES? ";N
20 FOR I=1 TO N
: MODE(I)=0
: NEXT
30 FOR I=1 TO N
40 PRINT "NAME OF VARIABLE ";I;" ($ FOR STRING)"
50 INPUT V$(I)
60 IF RIGHT$(V$(I),1)="$" THEN MODE(I)=3
70 INPUT "WIDTH OF FIELD? ";W$(I)
80 IF MODE(I)=3 THEN 100
90 INPUT "DECIMAL DIGITS? ";D$(I)
100 INPUT "HEADING 1? ";P1$(I)
110 IF LEN(P1$(I))>W$(I) GOTO 100
120 INPUT "HEADING 2? ";P2$(I)
130 IF LEN(P2$(I))>W$(I) GOTO 120
140 INPUT "HEADING 3? ";P3$(I)
150 IF LEN(P3$(I))>W$(I) GOTO 140
160 NEXT I
170 INPUT "STARTING PROGRAM LINE? ";PR
180 INPUT "INCREMENT FOR PROGRAM? ";INC
190 INPUT "SPACE BETWEEN COLUMNS? ";SP
200 DIM L$(100)
210 PLC=PR
: J=1
220 L$(J)=STR$(PLC)+" ?"
230 GOSUB 2120
240 T=0

```

Listing 6 continued on page 360

**This little
baby will
shake up the
computer
industry.**



Actual board size: 5 1/4 in. x 8 3/4 in. (14.8cm x 22.1cm)

**PROTEUS™ - the very first 16-bit
single board microcomputer.**

PROTEUS leads the way into a new generation of smaller, faster computers.

PROTEUS, from Innovative Electronics Technology, features a 16-bit modular processing unit designed to mount on the side of a 5 1/4" mini Winchester hard disk. It contains a Z8001 segmented CPU with 16K of ROM, 128K of dynamic RAM for user memory, 8K of static RAM for system memory, two serial ports and a floating-point processor.

Additional memory management boards can expand capacity to 48 Megabytes per module! And, our innovative design allows up to 256 PROTEUS boards to be linked to provide simultaneous access to 512 users without any degradation in speed!

The computer market will never be the same again.

For additional information and specifications, write:

**Innovative
Electronics
Technology Ltd.**

P.O. Box 5188
Vancouver, B.C.
Canada V6B 4B5

Tel: (604) 430-5188
Telex: 04-0368551 IOTAVCR

Listing 6 continued:

```

250 FOR I=1 TO N
260 T=T+W%(I-1)+SP
270 L$(J)=STR$(PLC)+" ? TAB("STR$(INT(T+(W%(I)-LEN(P1$(I)))/2+1))+");"+CHR$(
34)+P1$(I)+CHR$(34)+";"
280 GOSUB 2120
290 NEXT I
300 L$(J)=STR$(PLC)+" ?"
310 GOSUB 2120
320 T=0
330 FOR I=1 TO N
340 T=T+W%(I-1)+SP
350 L$(J)=STR$(PLC)+" ? TAB("STR$(INT(T+(W%(I)-LEN(P2$(I)))/2+1))+");"+CHR$(
34)+P2$(I)+CHR$(34)+";"
360 GOSUB 2120
370 NEXT I
380 L$(J)=STR$(PLC)+" ?"
390 GOSUB 2120
400 T=0
410 FOR I=1 TO N
420 T=T+W%(I-1)+SP
430 L$(J)=STR$(PLC)+" ? TAB("STR$(INT(T+(W%(I)-LEN(P3$(I)))/2+1))+");"+CHR$(
34)+P3$(I)+CHR$(34)+";"
440 GOSUB 2120
450 NEXT I
460 L$(J)=STR$(PLC)+" ?"
470 GOSUB 2120
480 L$(J)=STR$(PLC)+" FOR I = 1 TO M"
490 GOSUB 2120
495 T=0
500 FOR I=1 TO N
510 IF MODE(I)=3 THEN L$(J)=STR$(PLC)+" A$="+V$(I)+"(I)"
: GOSUB 2120
: GOTO 585
520 L$(J)=STR$(PLC)+" A="+V$(I)+"(I)"
525 GOSUB 2120
550 L$(J)=STR$(PLC)+" W$="+STR$(W$(I))+": D$="+STR$(D$(I))
560 GOSUB 2120
570 L$(J)=STR$(PLC)+" GOSUB 60000"
580 GOSUB 2120
585 T=T+W%(I-1)+SP
590 L$(J)=STR$(PLC)+" ? TAB("STR$(INT(T+W%(I)+1))+"-LEN(A$)); A$;"
595 GOSUB 2120
600 NEXT I
620 L$(J)=STR$(PLC)+" ?"
630 GOSUB 2120
640 L$(J)=STR$(PLC)+" NEXT I"
650 GOSUB 2120
2010 PRINT
: PRINT
2020 FOR K=1 TO J
: PRINT L$(K)
: NEXT
2030 INPUT "DO YOU WANT TO SAVE ON DISK?";Z$
2040 IF Z$<>"Y" THEN END
2050 INPUT "TEXT FILE NAME? ";F$
2060 D$=CHR$(4)
2070 PRINT D$;"OPEN";F$
2080 PRINT D$;"WRITE";F$
2090 FOR K=1 TO J
: PRINT L$(K)
: NEXT K
2100 PRINT D$;"CLOSE";F$
2110 END
2120 PLC=PLC+INC
: J=J+1
: RETURN

CREATE INSTR
10 HOME
20 INPUT "APPROXIMATELY HOW MANY LINES? ";I
30 DIM S$(INT(I*1.5))
40 D$=CHR$(4)
50 EQ$=CHR$(34)
60 CR$=CHR$(13)
70 BS$=CHR$(8)
80 QQ$=CHR$(17)

```

Listing 6 continued:

```

90 HAK$=CHR$(21)
100 PRINT "TYPE 'CONTROL-Q' TO QUIT"
110 PRINT "ANSWER QUESTIONS WITH 'Y' OR 'N'"
120 LN=1
130 REM
140 PRINT
150 PRINT "TYPE LINE ";LN
160 GOSUB 640
170 IF CH$<>QQ$ THEN GOTO 140
180 NL=LN-1
190 PRINT
: PRINT
200 FOR I=1 TO NL
210 PRINT S$(I)
220 NEXT I
230 PRINT
240 INPUT "DO YOU WANT TO CHARGE A LINE? ";Z$
250 IF Z$<>"Y" GOTO 360
260 INPUT "WHAT LINE? ";LN
270 IF LN>NL OR LN<1 GOTO 260
280 PRINT S$(LN)
290 PRINT
300 INPUT "IS THIS THE RIGHT LINE? ";Z$
310 IF Z$<>"Y" GOTO 260
320 S$(LN)=" "
330 PRINT "TYPE LINE ";LN
340 GOSUB 640
350 GOTO 190
360 INPUT "STARTING PROGRAM LINE? ";PLC
370 INPUT "INCREMENT FOR PROGRAM? ";INC
380 FOR I=1 TO NL
390 L=LEN(S$(I))
400 FOR J=1 TO L
410 IF L=0 THEN S$(I)=STR$(PLC)+"?"
: GOTO 480
420 IF LEFT$(S$(I),1)<>" " GOTO 450
430 S$(I)=RIGHT$(S$(I),LEN(S$(I))-1)
440 NEXT J
450 S1$="TAB("
: S2$=")";"
: S3$=STR$(J)
460 IF J=1 THEN S1$=""
: S2$=""
: S3$=""
470 S$(I)=STR$(PLC)+"?"*S1$+S3$+S2$+EQ$+S$(I)+EQ$
480 PLC=PLC+INC
490 NEXT I
500 FOR I=1 TO NL
510 PRINT S$(I)
520 NEXT I
530 PRINT
540 INPUT "DO YOU WANT TO SAVE ON DISK? ";Z$
550 IF Z$<>"Y" THEN END
560 INPUT "TEXT FILE NAME ";F$
570 PRINT D$;"OPEN";F$
580 PRINT D$;"WRITE";F$
590 FOR I=1 TO NL
600 PRINT S$(I)
610 NEXT I
620 PRINT D$;"CLOSE";F$
630 END
640 GET CH$
650 IF CH$<>CR$ AND CH$<>BS$ AND CH$<>QQ$ AND CH$<>HAK$ THEN PRINT CH$;
: S$(LN)=S$(LN)+CH$
: GOTO 640
660 IF CH$=BS$ AND LEN(S$(LN))<=1 THEN S$(LN)=" "
: HTAB 1
: GOTO 640
670 IF CH$=CR$ THEN PRINT CH$;
: S$(LN)=LEFT$(S$(LN),LEN(S$(LN))-1)
: GOTO 640
680 IF CH$=HAK$ THEN CH$="?"
: GOTO 650
690 IF CH$=CR$ THEN LN=LN+1
: RETURN
700 IF CH$=QQ$ THEN RETURN
710 STOP

```

Text continued from page 354:

L\$(J) is the concatenation of a number of substrings:

```

STR$(PLC)
" INPUT "
CHR$(34)
P$(I)
" "
CHR$(34)
" ; "
V$(I)
"(I)"

```

These substrings form INPUT statements, such as line 1050 in listing 1:

```
1050 INPUT "UNIT PRICE "; PR(I)
```

STR\$(PLC) generates the current statement number, 1050; " INPUT " generates the INPUT token; CHR\$(34) is the quote mark, " ; P\$(I) is the string for the prompt string of the Ith variable, in this case UNIT PRICE; and " " adds a space after PRICE. The trailing quote is then added. Next, a semicolon is placed in the string. Finally, the variable name for the Ith variable is inserted, followed by the subscript index, (I). Remember that the I in V\$(I) is completely different from the I in "(I)": the first I is the index for the Ith variable in the CREATE INPUT program; the second I is the index for the Ith item in the invoice program.

The first step in creating your own automatic program generators is to decide which parts of your programs *can* be generated automatically. Sections that are easily parameterized are prime candidates. Next, you must be able to write the program yourself. Once you do this, break the program down into those parts that are general and those that are to be customized. Create an interactive entry program (using the programs shown in listing 6) to define the customized parts. Then, following the examples given here, write the statements that create the strings for each program statement. These three utilities allow you to write programs for yourself or friends, clients or customers, in very little time. Using these techniques, the invoice program takes about ten minutes to write. Which is all to say—let your computer do the programming! ■